



**UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH**

Departament d'Enginyeria Elèctrica

Tesis doctoral

**Desarrollo de herramientas constructivistas para la
publicación y personalización de contenidos de
cursos en línea**

Autor: Javier Garcia Fernandez

Director: Oriol Boix Aragonès

Barcelona, octubre de 2015

Resumen

Desde la aparición de internet, se han sucedido las experiencias para utilizar este entorno en la formación; con éxito desigual. Unas veces porque eran complicadas de utilizar, otras por el estado de la tecnología o porque no recibieron el soporte del público. Actualmente existen cursos en línea, campus virtuales (LMS), wikis, vídeo, MOOC o aplicaciones educativas que aprovechan los dispositivos personales (m-learning).

Los cursos en línea están estancados. El papel del alumno es pasivo respecto a su aprendizaje. En su mayoría, son herramientas de estudio individual, estáticas, que no fomentan la colaboración entre individuos, ni la construcción de conocimiento y no se adaptan a las características del alumno. Hace años se estudió la creación de sistemas de aprendizaje adaptables pero no resultó viable.

Este trabajo busca innovar en los cursos en línea. Se ha construido un conjunto de herramientas que dan al alumno un papel activo en la construcción de su aprendizaje y al formador le permiten crear cursos más dinámicos y atractivos.

Se ha trabajado en una plataforma de publicación que permita a formadores con conocimientos básicos de HTML implementar cursos en línea. El sistema se encarga de añadir los paneles de navegación, aplicar plantillas de estilos y añadir el código para generar elementos interactivos, fórmulas científicas, etc.

Pensando en el estudiante, se han creado unas herramientas con las que puede crear una capa de personalización de los contenidos del curso (subrayado, glosarios, resúmenes, notas y calculadora) y guardarlos en la nube. Esta caja de herramientas es el resultado de trasladar los recursos tradicionales de estudio a las TIC.

Todo ello empleando tecnologías estándar de internet que funcionan sobre cualquier dispositivo que disponga de un navegador web.

El resultado de este trabajo será accesible desde <http://recursos.citcea.upc.edu/llum/>

Abstract

Since early times in internet, there have been training experiences with different success. Sometimes because they were complicated to use, others for the state of technology, or because the public has not supported it. Currently there are online courses, virtual campus (LMS), wikis, video, MOOC or educational applications that use personal devices (m-learning).

Online courses have not evolved. The role of the student is passive about their learning. Most courses are individual and static study tools that do not encourage collaboration between individuals and cannot adapt to the characteristics of the student, nor help them build their knowledge. Time ago, researchers studied the creation of adaptive learning systems but it was not viable.

This work wants to innovate in online courses. It has built a set of tools that give students an active role in building their learning and let the trainer create more dynamic and attractive courses.

It has worked in a publishing platform that enables trainers with basic knowledge of HTML implement online courses. The system is responsible for adding navigation panels, apply style templates and add the code to create interactive elements, scientific formulas, etc.

Thinking in the students, have prepared tools to create a layer of customization of course content (underlined, glossaries, summaries, notes and a calculator) and save changes in the cloud. This toolkit transforms the student traditional tools into web applications.

The entire project works with standard Internet technologies that work on any device with a web browser.

The result of this work will be accessible from <http://recursos.citcea.upc.edu/llum/>

Agradecimientos

Ahora que estoy tan cerca de terminar esta tesis después de tantos años durante los cuales han pasado muchas cosas en mi vida, tengo la sensación de que me queda un gran vacío. Decidí realizar el doctorado por unas circunstancias un tanto rocambolescas que no vienen al caso. Aquel día cuando volví a casa, les dije a mis padres que iba a completar mis estudios universitarios y se pusieron muy contentos. Mi padre estaba muy enfermo y se alegró mucho. Desgraciadamente no podrá ver como he llegado al final del camino, la enfermedad se lo llevó. Durante estos años han pasado muchas cosas: algunas buenas y otras malas, algunas incluso me impidieron dedicarme durante un tiempo a este proyecto. Afortunadamente, a periodos difíciles le siguen otros mejores y ahora estoy en uno de estos.

Quiero agradecerle a mi madre todos estos años de cariño y apoyo aunque muchas veces no supiera ni entendiera que hacía exactamente o porque no dedicaba más tiempo a la tesis. A mi hermana Montse por ser tan especial como es. A mi hermana Gloria, a su marido Carlos y mi sobrino Carles. A mi hermano Daniel y su mujer Kelly. A mi hermana Rocío y a su marido Antonio. Y a mi hermana Sonia, su marido Daniel y mis sobrinos María y Daniel. Y un especial recuerdo a mi padre que luchó por sacar adelante a su familia en unas condiciones a veces difíciles y que quiso dar la oportunidad a todos sus hijos de estudiar y prosperar en la vida.

También quiero acordarme de mi amigo Miguel Ángel al que conozco desde que éramos muy pequeños. Le agradezco todo el cariño, comprensión y amistad que me ha brindado estos años. Un recuerdo para su mujer Esperanza y su hija Laia. A su madre Ana que me ha tratado como un hijo más. A su hermano Iván y su mujer Carme. Y un recuerdo especial para su padre Miguel, un hombre emprendedor y familiar que nos dejó durante estos años.

Agradezco a Oriol Boix, mi tutor y amigo, al que conozco desde que hice al PFC su paciencia conmigo; por respetar mis tiempos que a veces son eternos. Su interés por innovar en temas docentes ha hecho posible que hiciera mi PFC y esta tesis. Muchas gracias de corazón.

Un recuerdo también para mis compañeros del CITCEA-UPC porque el tiempo que colaboré con ellos me permitió aprender e investigar muchas cosas que he podido aplicar en esta tesis y en mi vida laboral. En especial quiero dar las gracias a Antoni Sudrià por la confianza demostrada. Y muchas gracias a Samuel Galceran por su amistad y confianza durante todos estos años.

Muchas gracias a mis compañeros y personal del CIFO La Violeta por la oportunidad que me dieron de trabajar como formador. Allí he podido pulir mis conocimientos sobre tecnologías web y ayudar a gente a que tenga una formación que les facilite a encontrar trabajo en tiempos tan difíciles.

Quiero también acordarme y dedicarles también esta tesis a Yolanda y Namie que nunca creyeron que terminaría ni me apoyaron en esta empresa.

Finalmente quiero dar las gracias a María José por su ayuda y amistad para terminar esta tesis; has sido un gran apoyo estos meses últimos para acabar la tesis.

Y muy especialmente a Shirka, ejemplo de lucha contra la adversidad, gracias por tus ánimos y amistad. Este trabajo debe mucho a mi colaboración contigo en proyectos web que me obligó a superarme más. Tienes un universo propio muy rico que reflejas en tu arte. Dentro de poco ya podrás llamarme "doc" como en regreso al futuro. :P

Barcelona, octubre de 2015

Tabla de contenido

1	Introducción	11
2	Aspectos pedagógicos	15
3	La formación en línea	19
4	El medio	25
4.1	Internet.....	25
4.2	World Wide Web (WWW)	27
4.2.1	Arquitectura cliente servidor	28
4.2.2	El protocolo HTTP.....	30
4.3	Accesibilidad.....	31
4.3.1	Barreras debidas a las características de los dispositivos de los usuarios	32
4.3.2	Barreras debidas a las características de los navegadores	32
4.3.3	Barreras para las personas.....	33
4.4	Paradigmas de programación aplicables.....	36
4.4.1	Promoción de la web	36
4.4.2	Diseño no intrusivo	37
4.4.3	MVC.....	37
4.4.4	Programación orientada a objetos	40
4.5	Aspectos legales	42
4.5.1	Propiedad intelectual y licencia de uso.....	42
4.5.2	Tratamiento de datos personales	44
5	La tecnología	47
5.1	Criterios de selección	47
5.2	Selección tecnológica	49
5.3	Contenidos de la web	50
5.3.1	Lenguaje de etiquetas.....	50
5.3.2	Estilo de los documentos	52
5.3.3	Accesibilidad	53
5.3.4	Posicionamiento en buscadores	54
5.3.5	Alternativas.....	56
5.4	Formatos de imágenes	56
5.4.1	Mapa de bits o <i>raster</i>	56
5.4.2	Vectorial.....	58
5.5	Tecnologías transversales	59
5.5.1	Codificación de caracteres	59
5.5.2	Expresiones regulares	62

5.6	Intercambio de información entre el cliente y el servidor	62
5.6.1	JSON	63
5.6.2	XML	64
5.6.3	Alternativas	65
5.7	Persistencia de la información	66
5.7.1	Almacenamiento en el cliente	66
5.7.2	Almacenamiento en el servidor	68
5.8	Modificación dinámica del contenido de las páginas en el cliente.	70
5.9	<i>Server Side</i> y servidor de aplicaciones	71
5.10	Servidor web	74
6	Curso de pruebas	77
6.1	Planteamiento	77
6.2	Estructura del contenido del curso	78
6.2.1	La visión	79
6.2.2	La luz	79
6.2.3	Fotometría	79
6.2.4	Lámparas y luminarias	80
6.2.5	Iluminación de interiores	80
6.2.6	Iluminación de exteriores	80
6.3	Estructura del contenido de las páginas	81
6.4	Mapa web del curso	82
7	Publicación de los cursos	83
7.1	Plataforma de publicación	83
7.1.1	Objetivo	83
7.1.2	Funcionamiento	84
7.1.3	Programación	86
7.1.4	Estructura de carpetas del programa	91
7.1.5	Uso	92
7.1.6	Desarrollos futuros	92
7.2	Scripts para los contenidos multimedia	93
7.2.1	Fórmulas matemáticas en documentos hipermedia	93
7.2.2	Complementos educativos	96
8	Herramientas del estudiante	111
8.1	Arquitectura de la aplicación	111
8.2	Base de datos	122
8.3	Barra de herramientas y gestión de usuarios	124
8.4	Herramienta calculadora	130

8.4.1	Objetivo.....	131
8.4.2	La interfaz de usuario.....	132
8.4.3	Ayuda	134
8.4.4	Funcionamiento	134
8.4.5	Implementación	135
8.5	Herramienta notas	137
8.6	Herramienta glosario.....	141
8.7	Herramienta resumen	142
8.8	Herramienta subrayado	144
8.9	Herramienta portafolio	148
9	Conclusiones	151
10	Desarrollos futuros.....	153
11	Bibliografía	157
12	Índice de ilustraciones	167
13	Índice de fragmentos de código.....	169
14	Índice de tablas	171

1 Introducción

Corría el año 1997 cuando estaba terminando mis estudios universitarios. Como el resto de mis compañeros buscaba un tema para desarrollar el proyecto final de carrera necesario para obtener el título cuando, por aquellas casualidades de la vida, acabé interesándome por un anuncio en que buscaban alumnos que quisieran realizar proyectos relacionados con la docencia por Internet.

En aquella época, Internet aún no era esa cosa omnipresente que abarca casi todos los ámbitos de nuestra vida y sin la cual no parece que se pueda vivir. Era un terreno virgen en el que todo estaba por hacer. Yo vivía aquello con mucha emoción. Había tenido el privilegio de ver el nacimiento de una revolución tecnológica tan importante como la industrial que se produjo en el siglo XIX.

Quedé con mi futuro tutor, Oriol Boix, profesor del departamento *d'Enginyeria Elèctrica de la UPC* con el que mantengo una gran relación desde entonces. De hecho, también es el tutor de esta tesis. Es una persona que siempre se ha destacado por su interés en innovar y aplicar las nuevas tecnologías a la docencia. Estuvimos hablando sobre qué podía hacer y me explicó que su idea era crear un curso en línea; un manual de autoaprendizaje que sirviera para estudiar las posibilidades de este nuevo medio. Hasta entonces, él había participado en la creación de materiales multimedia en soporte de CD.

Le presenté una propuesta para elaborar un curso de luminotecnia en formato HTML, la aprobó y me lancé a la doble tarea de redactar los contenidos y maquetarlos sin ningún conocimiento previo del medio. Era necesario pensar cómo se iban a organizar los contenidos, el sistema de navegación, resolver cómo mostrar las fórmulas científicas y cómo añadir interactividad al curso. Fue un trabajo intenso al que me entregué con mucha pasión. La prueba es que a día de hoy continúo trabajando en él y haciendo mejoras como las que se presentan en esta tesis¹.

A pesar de lo innovador de la temática del proyecto, no tuvo una buena acogida por parte de los responsables de proyectos de la escuela. No supieron ver las posibilidades que el medio podía aportar a la docencia. Resulta paradójico cuando en años posteriores se han presentado al menos otros dos proyectos de manuales de luminotecnia y juno de ellos por Internet!²

Tal como consta en la memoria del PFC: "El material elaborado en este proyecto pretende demostrar las posibilidades reales de utilizar Internet como herramienta de enseñanza, descubrir sus limitaciones y las posibilidades actuales y futuras de desarrollo de herramientas o métodos de diseño e implementación sobre los que apoyar nuevos proyectos docentes a llevar a cabo en el futuro por el departamento."

El resultado fue francamente bueno y con los años, ha pasado a ser en un material de referencia en el mundo de la luminotecnia en todos los niveles y ámbitos de formación³.

Tras esta experiencia me incorporé al mundo laboral y me dediqué a la elaboración de páginas web. Durante este tiempo, programé aplicaciones de cliente y servidor web. Incluso realicé algunos cursos para una conocida empresa de formación a distancia.

Seguí formándome y adquiriendo más y más experiencia en programación web y tras unos años de colaboración con un centro de investigación de la universidad, terminé dedicándome a la formación de desempleados en el área de bases de datos, programación y desarrollo de aplicaciones web en entornos cliente y servidor; trabajo al que continúo dedicándome.

¹ <http://recursos.citcea.upc.edu/llum/>

² <http://grlum.dpe.upc.edu/manual/>

³ Basta en introducir en Google la siguiente búsqueda para obtener cientos de resultados: <http://edison.upc.edu/curs/llum/>

Toda esta experiencia laboral y la preparación de los cursos me han permitido alcanzar un alto nivel de conocimiento de las tecnologías web que ha resultado imprescindible para desarrollar este proyecto. Las TIC⁴ son un sector en constante evolución, en el que es necesario que los profesionales se estén reciclando continuamente para no quedar desfasados. La principal fuente de conocimiento son los manuales de referencia y las guías publicadas por los desarrolladores de *software* y lenguajes de programación, las páginas con tutoriales, artículos de divulgación tecnológica, páginas de ayuda a desarrolladores, etc.; en general, cualquier publicación de donde se puedan extraer ideas válidas.

Mi experiencia como formador, me ha permitido interactuar con muchos alumnos y conocer sus necesidades y dificultades educativas. Muchos de ellos han aprendido de forma autodidacta con mejor o peor suerte. En general, se aprecia que mientras mejor sea su formación de base o experiencia laboral, más solventes son. Otros, neófitos o con carencias de conocimientos, aprecian las clases presenciales donde el experto⁵, que previamente ha realizado una labor de estudio, síntesis y elaboración del material del curso, les enseña, guía y prepara para que adquieran las competencias profesionales objeto de la formación. Dos de las competencias que considero más importantes son que los alumnos adquieran la capacidad de buscar recursos formativos y utilizarlos y que dispongan de criterios para discernir si un recurso les resultará útil o no.

Porque cuando uno consulta recursos en Internet, se da cuenta de dos cosas: una, que los manuales son tan extensos que no es práctico imprimirlos; y dos, que es complicado personalizar el contenido con acciones como anotaciones, subrayados... que son tan fáciles de hacer en el papel. En algunos formatos como PDF o documentos de ofimática (texto, hojas de cálculo...) esto es posible, pero en HTML no⁶. Y esto era una cosa que echaba de menos en el curso de iluminación.

Para el autor, Internet ofrecía muchas más posibilidades de las ofrecidas en el curso. Es cierto que el contenido está bien estructurado, con textos concisos que ofrecen diferentes niveles de conocimientos y contenidos interactivos. Pero aunque, ni mis conocimientos en aquel entonces ni la tecnología existente permitían mucho más, siempre me quedé con la inquietud de no haber cumplido totalmente el objetivo que me había marcado.

Durante estos casi quince años de existencia del curso lo he ido renovando y adaptando a las nuevas tecnologías, buscando que fuera más práctico y útil como herramienta docente.

Me ha interesado especialmente la creación de interfaces de usuario que permitan plasmar ricas aplicaciones y sirvan para transformar los cursos en entornos realmente innovadores. Porque siempre pensé que los recursos docentes en Internet adolecían del mismo mal: ser una conversión de libros en páginas HTML. En mi opinión, se pierde el verdadero sentido del medio: el hipertexto, las interfaces complejas, las aplicaciones que den valor añadido a los servicios web... en definitiva, la posibilidad de que el medio se adapte al usuario. En los últimos años, gracias a la irrupción de nuevos dispositivos, teléfonos inteligentes y tabletas, han nacido nuevas maneras de interacción entre los usuarios e Internet. A los medios de estudio tradicionales, le han salido nuevos dispositivos que permiten al estudiante centrarse más en aprender y comprender las materias y menos en memorizar contenidos.

Por tanto esta tesis tiene el modesto interés de demostrar que es posible concebir los cursos en línea de una manera diferente. Que no hay que conformarse con hacer un libro, sino que pueden crearse herramientas que permitan al alumno interactuar con los contenidos y adaptarlos a sus necesidades.

Para conseguir este objetivo se ha trabajado en dos frentes. Por un lado se han creado herramientas, pensando en los formadores, que facilitan la producción de los cursos y la inclusión de elementos interactivos y multimedia. Su uso es

⁴ Tecnologías de la información y la comunicación.

⁵ En la terminología del SOC (*Servei d'Ocupació de Catalunya*) los formadores se conocen como expertos.

⁶ Microsoft Edge®, aparecido a finales de julio de 2015 y sucesor de Microsoft Internet Explorer®, que se incluye en Microsoft Windows® 10 es el primer navegador popular que incorpora la posibilidad de hacer anotaciones y subrayados. Su aparición es posterior al desarrollo de esta tesis.

muy sencillo, ya que se han ideado como cajas negras que se utilizan sin necesidad de conocer cómo funcionan internamente.

Y por otro, pensando en el alumnado, se han creado unos instrumentos, las *Web Students Tools* o barra de herramientas del estudiante, que permiten crear una capa de personalización en el curso. De esta manera, el alumno puede modificar el contenido de los cursos y adaptarlo a sus necesidades sin perder el acceso al original. Las herramientas se han concebido como una traslación de las técnicas y utensilios de estudio tradicionales del alumno (resúmenes, anotaciones, glosario, calculadora...) al campo de las TIC. Aunque en origen se han pensado como un complemento para la formación en línea, no hay ningún impedimento para que empleen en cualquier página web; pues se han diseñado para ser autónomas de la página donde se ejecutan.

La realización del trabajo se ha estructurado en diferentes fases. En primer lugar, se han analizado todos los elementos a tener en cuenta en el desarrollo de proyectos web, en especial los tecnológicos. A continuación, se han definido los criterios para seleccionar las tecnologías que se utilizarán en la implementación de las herramientas. Después, se ha definido una arquitectura a seguir en la programación de los instrumentos para la publicación los cursos y crear elementos interactivos. De igual manera, se ha especificado la arquitectura para la creación de las herramientas del estudiante. Finalmente, se ha implementado todo en un curso de pruebas donde se pueden ver los resultados del trabajo.

2 Aspectos pedagógicos

El viejo paradigma de que todo lo necesario para conocer un tema está en un libro o en unas clases magistrales ha quedado atrás en buena medida. En cualquier disciplina, la evolución y creación de nuevo conocimiento es constante e internet se encarga de su rápida difusión, que sirve a su vez de base para una nueva generación de ideas que reinician el ciclo.

Para los trabajadores del mundo moderno, la formación continua es un activo de su carrera profesional. Es una forma de distinguirse del resto de personas y de ganar competitividad. Tener un perfil diferenciado del resto es importante. La experiencia laboral es una excelente fuente de conocimientos, pero a menudo, esta relación empresa-empleado impide a este último investigar o probar en terrenos diferentes de los marcados por la entidad donde se trabaja.

En el caso de los alumnos de estudios medios y superiores, la formación reglada les ofrece la oportunidad de desarrollar una buena base de conocimientos que les permita, en el futuro, adquirir otros nuevos por su propia cuenta. "Aprender a aprender" es una de las competencias más importantes que hay que desarrollar en esta etapa de la vida.

"[...] Aprender a aprender significa que los estudiantes se comprometan a construir su conocimiento a partir de sus aprendizajes y experiencias vitales anteriores con el fin reutilizar y aplicar el conocimiento y las habilidades en una variedad de contextos: en casa, en el trabajo, en la educación y la instrucción.[...]" [1]

En la sociedad de internet, donde hay grandes cantidades de conocimientos fácilmente accesibles en tiempo, espacio y a bajo coste, es necesario que el alumno desarrolle competencias que le ayuden a sobrevivir ante esta avalancha de información. Debe aprender tres cosas: a buscar fuentes de información utilizando los múltiples medios y dispositivos a su disposición (ordenadores, teléfonos inteligentes, prensa, televisión...), a discernir la calidad, validez y veracidad de las fuentes y la utilidad de la información obtenida para el propósito fijado (no tiene sentido disponer de una información que no se es capaz de comprender).

El formador ha de ser consciente de que cada alumno es un mundo. Cada persona tiene un estilo y un ritmo propios de aprendizaje. En la formación de adultos se nota especialmente que la experiencia vital y los conocimientos previos son fundamentales para un buen proceso de aprendizaje. Son personas que, en general, tienen las ideas claras; se han marcado unos objetivos y buscan una formación adecuada que les ayude a alcanzarlos.

En la formación de adultos, se puede concluir que existen dos tipos de alumnos: los que se introducen en una nueva disciplina y carecen por tanto de conocimientos previos y los que sí los tienen. Para los primeros, empezar la formación es complicado porque no tienen una base con la que asociar los nuevos contenidos. Por eso es importante que las explicaciones sean más sencillas y los ejemplos incluyan cosas de la vida cotidiana que puedan relacionar. En el segundo tipo, todo es más fácil, puesto que asocian lo que aprenden a su experiencia previa en el tema y lo comprenden con más facilidad. Por eso, avanzan más rápidamente y, en general, con mejores resultados. Una vez se dispone de una base suficiente en una materia, es factible acceder y asimilar información más compleja, desarrollar capacidades para buscar nuevos conocimientos y disponer de criterio para discernir su calidad y utilidad.

Mi experiencia como formador en el campo de las TIC me proporciona un buen conocimiento del medio, sus posibilidades tecnológicas y las dificultades de crear recursos formativos en línea adaptados a todos los tipos de alumnos. Por otra parte, cuando se plantea un curso hay que decidir a qué audiencia irá dirigido, si habrá prerequisites para el estudiante (conocimientos previos requeridos), cómo se seguirá el curso, diferenciar distintos niveles de profundización en el aprendizaje, incorporar interactividad y ejemplos que ayuden a asociar el conocimiento expuesto a situaciones de la vida real... A menudo, conjugar todos estos elementos no es posible y habrá que llegar a soluciones de compromiso.

Este trabajo se compone de una plataforma de publicación para cursos y la barra de herramientas del estudiante, *Web Student Tools*. Para probar las tecnologías desarrolladas se ha empleado, como muestra, un curso de luminotecnia. El

curso, que fue desarrollado por el autor para su proyecto final de carrera, se ha sometido a una puesta al día formal y tecnológica para integrarlo en las nuevas herramientas.

El curso de luminotecnia versa sobre la disciplina de la iluminación, desde la luz y la visión hasta como realizar cálculos de iluminación. Está dirigido principalmente a alumnos de enseñanza superior, aunque se ha buscado también que tenga un carácter divulgativo. Las explicaciones se presentan de forma progresiva y ordenada, para que sea fácil asociar un tema con los precedentes. De esta manera, se facilita que los alumnos sin experiencia en la materia puedan seguir el curso. Los alumnos más avanzados, sin embargo, pueden optar por navegar sin un orden establecido, haciendo uso del acceso directo al tema o temas de su interés.

Las unidades del curso se dividen en temas, redactados para ofrecer un nivel de conocimientos básicos esenciales para su comprensión por el alumno y que a la vez les resulten prácticos. Paralelamente, existe otro nivel más avanzado en el que se demuestran las fórmulas y se explican conceptos complementarios al tema (pero no esenciales). Además, siempre que ha sido posible, se han incorporado animaciones o elementos interactivos que hagan más activo el papel del alumno. Por último, los temas donde se explican técnicas de cálculo se acompañan de ejercicios para el estudio y autoevaluación del alumno con sus correspondientes soluciones.

En resumen, el curso se ha pensado como una herramienta de autoformación de fácil acceso y uso que el usuario pueda seguir a su ritmo según sus conocimientos previos en el tema y su situación personal.

El siguiente elemento es la plataforma de publicación. Sirve de soporte a la difusión del material compuesto por el formador, sin que éste tenga que preocuparse por los aspectos más técnicos como pueden ser el aspecto, la creación de elementos interactivos, las fórmulas matemáticas o el sistema de navegación del curso del que se ocupa el sistema. De esta manera el formador sólo ha de preocuparse de la redacción de contenidos, creación de imágenes, etc. Así se facilita la producción de materiales de enseñanza por parte del formador y se consigue un medio en el que el papel del alumno sea más activo.

Por último, las herramientas del estudiante, motivan al alumno para que construya su conocimiento, enriqueciendo su estudio añadiendo comentarios y anotaciones, que puede compartir con el resto de compañeros e incorporar a otros documentos a través de la herramienta del portafolio. De esta manera, el usuario construye su propio conocimiento y se estimula que participe activamente de su proceso de formación.

Las diferentes herramientas permiten al usuario establecer y decidir qué criterios considera válidos para el aprendizaje de la materia. Ayuda al alumno a relacionar las ideas y estructurarlas mentalmente. La ventaja es que puede plasmar estas ideas con una única herramienta y no mantenerlas dispersas, como ocurre en la actualidad (diferentes páginas web, diferentes manuales, cálculos que realizaba con su calculadora física...).

Desde un punto de vista pedagógico, este trabajo contiene elementos que pueden circunscribirse en la teoría constructivista y en la conectivista.

El constructivismo es una teoría pedagógica que afirma que el aprendizaje es un proceso en el que el alumno crea nuevo conocimiento y aprende con mayor efectividad si se apoya en experiencias y enseñanzas actuales y pasadas. El alumno se convierte en un actor activo de su aprendizaje y la labor del formador es ayudarle creando un clima favorable y dándole las herramientas necesarias para resolver los problemas que se le plantean. [2], [3]

"La idea central es que el aprendizaje humano se construye, que la mente de las personas elabora nuevos conocimientos a partir de la base de enseñanzas anteriores. El aprendizaje de los estudiantes debe ser activo, deben participar en actividades en lugar de permanecer de manera pasiva observando lo que se les explica.

El constructivismo difiere con otros puntos de vista, en los que el aprendizaje se forja a través del paso de información entre personas (maestro-alumno), en este caso construir no es lo importante, sino recibir. En el constructivismo el aprendizaje es activo, no pasivo. Una suposición básica es que las personas aprenden cuándo

pueden controlar su aprendizaje y están al corriente del control que poseen. Esta teoría es del aprendizaje, no una descripción de cómo enseñar. Los alumnos construyen conocimientos por sí mismos." [4]

El planteamiento del curso tiene un enfoque constructivista pues se busca un avance progresivo en la materia en el que el alumno pueda interactuar y construir un nuevo conocimiento gracias al uso de las herramientas del estudiante.

El conectivismo es una teoría que defiende que el conocimiento se forma a través de establecer redes y conexiones entre las personas y/o otras fuentes de información formal o informal (libros, internet...). Lo importante es la red en sí misma, pues a través de ella podemos acceder a los conocimientos. En este contexto, el proceso de toma de decisiones sobre qué aprender y el significado de la información es también un proceso de aprendizaje. Es una teoría muy influenciada por el uso de la tecnología en la enseñanza. [5], [6], [7]

"Principios del conectivismo:

- El aprendizaje y el conocimiento dependen de la diversidad de opiniones.
- El aprendizaje es un proceso de conectar nodos o fuentes de información especializados.
- El aprendizaje puede residir en dispositivos no humanos.
- La capacidad de saber más es más crítica que aquello que se sabe en un momento dado.
- La alimentación y mantenimiento de las conexiones es necesaria para facilitar el aprendizaje continuo.
- La habilidad de ver conexiones entre áreas, ideas y conceptos es una habilidad clave.
- La actualización (conocimiento preciso y actual) es la intención de todas las actividades conectivistas de aprendizaje.
- La toma de decisiones es, en sí misma, un proceso de aprendizaje. El acto de escoger qué aprender y el significado de la información que se recibe, es visto a través del lente de una realidad cambiante. Una decisión correcta hoy, puede estar equivocada mañana debido a alteraciones en el entorno informativo que afecta la decisión." [6]

El curso de luminotecnia está concebido como una fuente de información sobre el tema. Con los conocimientos adquiridos en él, el alumno puede construir una base que le permita buscar otras fuentes con las que ampliar su aprendizaje. Las herramientas del estudiante contribuyen a este objetivo, pues permiten la inclusión de enlaces a otras páginas en las notas, resúmenes y glosarios. La idea, en definitiva, es favorecer que el estudiante utilice el curso como una fuente de información sobre la luminotecnia, y a partir de ella y de los recursos que le ofrece las herramientas, construya y conecte la información obtenida, aprendiendo y creando conocimiento.

3 La formación en línea

El WWW⁷ nace en el CERN⁸ en 1990 como un medio de intercambio y publicación de información entre los investigadores del centro. Los documentos creados se enlazan entre ellos mediante hipertexto⁹ y forman así, una red de páginas a través de las cuales se puede navegar.

Se desarrolla en un ámbito científico y al principio las capacidades de publicación que ofrece son muy limitadas. El lenguaje empleado, el HTML, nace con sólo dieciocho etiquetas (*tags*) [8]; sin capacidad de incorporar elementos multimedia, ni estilos, ni interactividad, etc. Sólo permite escribir párrafos, poner la firma del autor, glosarios, listas, cabeceras, el título del documento, enlaces y poco más.

Con la aparición en 1993 del primer navegador gráfico MOSAIC¹⁰, el WWW se popularizó y se comenzaron a incorporar nuevas extensiones al lenguaje (nuevas etiquetas) que permitían incorporar imágenes, estilos, etc. La progresión continuó hasta la situación actual en que se puede hacer prácticamente de todo. Inmerso en una continua evolución, es el protagonista de una de las revoluciones tecnológicas más importantes de la historia.

En el ámbito de la formación, la evolución ha ido desde el principio, muy ligada a las capacidades del medio y la disponibilidad de herramientas para la publicación de contenidos. Tal es así, que se ha constituido en una auténtica barrera tecnológica. Es un error pensar que los cursos en línea son el único tipo de acción formativa que se puede desarrollar en internet. A medida que ha cambiado la tecnología, han ido apareciendo nuevas formas de aprendizaje como videos, audio, realidad aumentada... que han expandido el concepto de aprendizaje y han facilitado la producción y publicación de contenidos.

Es muy tentadora la idea de entender internet como una forma y/o evolución de la formación a distancia, pero eso es simplificar mucho el tema. Internet ofrece multitud de posibilidades. Aunque los alumnos y docentes no compartan el mismo espacio y el alumno deba realizar básicamente un trabajo personal, existen canales de comunicación más o menos inmediatos (el correo, los chats o incluso las videoconferencias) que permiten que éstos se interrelacionen. Con las últimas tecnologías disponibles, es posible también crear contenidos interactivos, simulaciones... que tanto alumnos como profesores pueden incorporar al proceso de aprendizaje.

En los primeros tiempos de internet, las únicas acciones formativas posibles consistían en trasladar el material (libros, cursos...) al nuevo medio, incluir enlaces a documentos complementarios o colgar directamente los libros en formatos como el PDF. De hecho, aún hay entornos en los que se continúa trabajando así. El hipertexto representaba un gran avance respecto a los libros de texto en papel, pero si no se hacía un esfuerzo por dotarlo de un sistema de navegación coherente, era fácil que el usuario acabara perdiéndose entre las páginas. En resumen, en su primera etapa proliferaban los contenidos estáticos con algunos elementos multimedia (imágenes) y alguna o ninguna interactividad que resultaban poco atractivos para los estudiantes. En general, tampoco se establecía ninguna relación entre el formador y el alumno.

Cuando la tecnología evolucionó, y se pasó de los documentos estáticos a los dinámicos generados *in situ* en el servidor¹¹, se abrió un nuevo mundo de posibilidades. La aparición de los CMS¹² (*Content Management System*) revolucionó la producción de contenidos pues ya no era necesario tener conocimientos de HTML para crear páginas. En

⁷ World Wide Web

⁸ Organización Europea de Investigación Nuclear. <http://home.web.cern.ch/>

⁹ Sistema que permite la relación de documentos o el acceso a partes de estos de forma no lineal saltando de uno a otro.

¹⁰ https://en.wikipedia.org/wiki/Mosaic_%28web_browser%29

¹¹ Mediante servidores de aplicaciones y lenguajes como ASP, PHP, JEE CGI...

¹² "Un sistema de gestión de contenidos (CMS) es una aplicación informática que permite publicar, editar y modificar el contenido, organizar, borrar, así como el mantenimiento desde una interfaz centralizada. [...] Los CMS existen desde finales de la década de 1990." traducido de [132]

este momento se popularizan los blogs que permiten a las personas compartir experiencias y aparecen las primeras redes sociales, los portales de videos, etc.

En el campo de la formación aparecen y se popularizan los primeros campus virtuales, CMS especializados que ofrecen una infraestructura para la publicación de cursos y la gestión de los alumnos. Constituyen una comunidad educativa donde se ofrecen todos los servicios requeridos por el alumno y la institución para impartir los cursos. Son los conocidos como LMS (*Learning Manage System*) y como ejemplos se pueden citar Moodle¹³, Claroline¹⁴, Blackboard, Sakai, Dokeos, .LRN, eFront... Con ellos, la institución educativa tiene a su disposición una herramienta para gestionar alumnos, conocer su actividad y progresos. Los alumnos disponen de un entorno centralizado donde pueden acceder a los servicios administrativos, comunicarse con otros miembros de la comunidad educativa, acceder a recursos, apuntes, entregar trabajos, etc. Y los formadores pueden seguir la actividad de los alumnos, contactar más fácilmente con ellos, distribuir el material del curso e incluso realizar pruebas automatizadas de evaluación. En muchos casos se convierten en un complemento para las instituciones educativas tradicionales. Un lugar donde ofrecer servicios al alumno y un repositorio de documentos. Se favorece la creación de una comunidad virtual y se ofrecen servicios extra como chats, foros de intercambio de información, correo electrónico, calendarios, evaluación, seguimiento del alumno, etc.

Con el tiempo y para facilitar el intercambio de contenidos entre las diferentes plataformas, la industria creó estándares como SCORM¹⁵ (*Shareable Content Object Reference Model*).

Aunque los LMS solucionan muchos de los problemas inherentes a la formación en internet, no suponían un especial avance en el desarrollo de cursos más atractivos y dinámicos para los estudiantes; ni facilitan el trabajo del docente en la creación de cursos. Sin embargo, hay que remarcar algunos aspectos positivos que sí ofrecen estas plataformas: fomentan el trabajo colaborativo y buscan la implicación y participación de los miembros de la comunidad en el proceso de aprendizaje [9].

Pero continuaba habiendo un problema: no se disponía de la tecnología necesaria para crear cursos. O bien se disponía de un repositorio de PDF y enlaces a páginas estáticas, o era necesario ser un manitas para crear cursos utilizando el, cada vez más, complicado y sofisticado lenguaje HTML. A todo esto, el papel del alumno en el curso seguía teniendo un fuerte componente pasivo, ya que se le entregaban unos contenidos cerrados con los que poco podía interactuar.

Pero, aunque se pueda pensar lo contrario, la falta de conocimientos y la barrera tecnológica no es una excusa para que no se desarrollen acciones formativas. Cuando aparece una dificultad, el ser humano siempre encuentra una brecha para superarla. Esto es así en la historia de la humanidad y de la ciencia. Si producir contenidos escritos y multimedia es complicado y requiere avanzados conocimientos, ahí está el vídeo y plataformas como YouTube¹⁶. En la actualidad, es muy sencillo producir vídeo y publicarlo con los teléfonos inteligentes (*Smartphone*). Con ello, se ha conseguido que una gran cantidad de personas compartan sus conocimientos. En pequeños vídeos, que tratan sobre temas muy concretos, mucha gente explica desde las cosas más banales (trucos de limpieza, bricolaje...) hasta clases magistrales y conferencias de profesionales de reconocido prestigio. No hay que depreciar estas acciones. Quizás no todas tengan la misma calidad; pero hay que medir su utilidad, si el mensaje llega al oyente, si sirve de algo lo que explican, y no tanto el valor de los medios.

El siguiente gran salto en el campo del *e-learning* fue la aparición de los MOOC (*Massive Open Online Course*).

"Los Cursos Masivos Abiertos en Red, o MOOC [...], son una nueva modalidad de formación con propuestas orientadas a la difusión web de contenidos y un plan de actividades de aprendizaje abierto a la colaboración y la participación masiva. Son cursos con soporte web escalable e inscripción libre para quienes quieran acceder y seguir la propuesta formativa." [10]

¹³ <https://moodle.org>

¹⁴ <http://www.claroline.net/>

¹⁵ <http://www.adlnet.org/scorm>

¹⁶ Canal de vídeo del director de la tesis: <https://www.youtube.com/user/oriolboixa>

Los MOOC aparecieron impulsados por grandes instituciones universitarias de los EUA para hacer accesible la educación superior en lugares donde no es posible o es difícil. Su popularización se produce a partir de 2012, gracias a que universidades de todo el mundo apuestan por ellos. Ofrecen cursos de temáticas muy concretas, sin restricciones de acceso, son gratuitos, tienen sistemas de evaluación de los progresos del alumno y emplean las redes sociales o foros para facilitar la interacción entre estudiantes y formadores. Aunque no tienen coste para los alumnos, es posible que tengan que pagar por obtener un certificado que acredite sus conocimientos. Los contenidos pueden consistir en videos con las lecciones, documentos colgados o, incluso, ser elaborados y reinterpretados por los participantes. Entre las críticas que reciben está el alto porcentaje de abandono, entre el 75 y el 95% [11], [12] y los sistemas de evaluación (en general de tipo cuantitativo y en formato test).

Las últimas tendencias en formación en línea se basan en el uso de los dispositivos personales móviles (teléfonos inteligentes, tabletas...), *m-learning*, como un recurso educativo más. Este terreno es más heterogéneo pues parte del uso de diferentes aplicaciones instaladas en los aparatos. Por ejemplo, imaginemos una aplicación de realidad aumentada¹⁷ que explique a los visitantes unas ruinas arqueológicas, la historia de los edificios y objetos hallados, y para ello tan solo deba enfocar con el móvil el objeto de su interés. E incluso que puedan ver una recreación de cómo era el lugar que están pisando en épocas pretéritas. En general, el concepto de *m-learning* se puede ampliar a cualquier dispositivo portátil de uso personal: reproductores de audio, ordenadores portátiles... Se trata pues de un medio pedagógico disperso e informal, pues las acciones formativas pueden desarrollarse en formatos muy heterogéneos como por ejemplo video, audio guías, etc.

"El m-learning [...] se define como "aprender a través de múltiples contextos, mediante interacciones sociales y de contenido, utilizando dispositivos electrónicos personales". Es una forma de educación a distancia, donde los alumnos utilizan tecnología educativa sobre dispositivos móviles para avanzar a su propio ritmo.

Las tecnologías m-learning incluyen ordenadores portátiles, reproductores MP3, *notebooks*, teléfonos móviles y tabletas. El m-aprendizaje se centra en la movilidad del alumno que interactúa con las tecnologías portátiles. El uso de herramientas móviles para la creación de materiales de aprendizaje se convierte en una parte importante del aprendizaje informal.

El m-learning es accesible desde prácticamente cualquier lugar. Los individuos que participan comparten de forma casi instantánea los mismos contenidos, lo que conduce a la recepción instantánea de la información y sugerencias. [...] El m-learning se caracteriza por una gran portabilidad que sustituyen los libros y cuadernos por pequeños dispositivos, llenos de contenidos de aprendizaje a medida." traducido de [13].

Otros campos de investigación en la formación por internet que se pueden mencionar aunque no hayan tenido tanta repercusión pública son los laboratorios de prácticas y la implementación de metodologías alternativas de aprendizaje. No han sido propuestas de éxito, muchas veces por las limitaciones tecnológicas del medio y otras por la dificultad de implementación.

Las prácticas de laboratorio son una actividad muy común en las enseñanzas técnicas y sirven para que el alumnado pueda relacionar los conocimientos adquiridos en las clases teóricas con un contexto real. Pueden ir desde realizar prácticas por ordenador hasta utilizar modelos a escala o máquinas reales. A pesar de que cada práctica tiene su finalidad y razón de ser, a menudo son percibidas como un trámite y son tratadas sin demasiado interés por estos. Además, los laboratorios de prácticas ocupan un valioso espacio que no se puede dedicar a la investigación y para el docente, en relación a los beneficios obtenidos, la inversión necesaria no resulta satisfactoria en general. Ese tiempo,

¹⁷ "La realidad aumentada (RA) es el término que se usa para definir una visión a través de un dispositivo tecnológico, directa o indirecta, de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real. Consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente, es decir, añadir una parte sintética virtual a lo real. Esta es la principal diferencia con la realidad virtual, puesto que no sustituye la realidad física, sino que sobreimprime los datos informáticos al mundo real." [71]

podría dedicarse a otras actividades más productivas como la investigación, la innovación pedagógica o la atención personalizada a los alumnos.

Con la tecnología existente y el empleo masivo de simulaciones, es posible cuestionarse la utilidad de realizar las prácticas de laboratorio en un entorno real. ¿Se pueden conseguir los mismos resultados con entornos virtuales? Actualmente, ya es común emplear simuladores de laboratorios¹⁸ que operan en las aulas informáticas en muchos ámbitos. El siguiente paso es el empleo de simuladores, laboratorios web, que funcionen desde internet.

La bibliografía del tema¹⁹ distingue dos tipos de laboratorios:

- Laboratorios web remotos. Están formados por una maqueta con el montaje, una cámara y un dispositivo y/o interfaz de control accesible vía web. Entre las principales limitaciones está la necesidad de disponer de un sistema de gestión de usuarios que limite el tiempo de acceso de cada uno. También es necesario disponer del espacio, la infraestructura, los equipos, el personal para el mantenimiento y un sistema de seguridad que evite daños en el equipo en caso de que los alumnos lo operen de forma incorrecta.

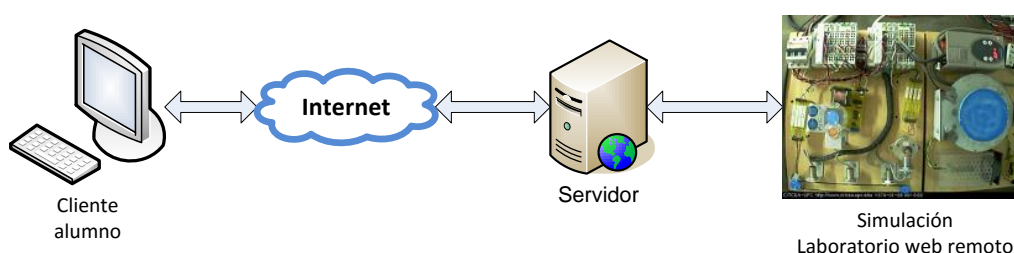


Ilustración 1 Laboratorio web remoto

- Laboratorios web virtuales. Estos laboratorios se caracterizan por su accesibilidad tanto en el tiempo como en el gran número de personas que se pueden conectar simultáneamente, y que las prácticas se simulan mediante un *software*.

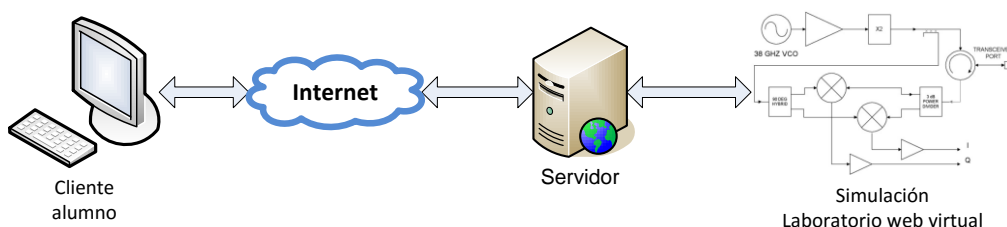


Ilustración 2 Laboratorio web virtual

El caso más interesante es el segundo. Así como durante estos años se han producido numerosos avances en formación por internet como los campus virtuales, los MOOC... en el campo de los laboratorios remotos virtuales no se ha pasado de una fase inicial donde abundan las experiencias aisladas²⁰, sin que haya sido capaz de desarrollarse una industria que lo impulse. Quizás la heterogeneidad de las materias (ingeniería, química, matemáticas, biología...), que provoca que los proyectos hayan de ser en buena medida realizados ex profeso, dificulte la creación de entornos estandarizados y reutilizables. Si a esto le añadimos que se necesitan de técnicos con elevados conocimientos para la programación de las simulaciones e interfaces y que hasta la llegada de HTML5 la tecnología web estaba muy limitada, no resulta el caldo

¹⁸ Se utiliza software científico de simulación como Simulink, Matlab o LabView.

¹⁹ [169], [144], [138]

²⁰ A modo de ejemplos, podemos citar:

<http://www.ilabcentral.org/>

<http://weblab.deusto.es/website/labs.html>

<http://web.mit.edu/edtech/casestudies/ilabs.html>

de cultivo más adecuado para su desarrollo. En opinión del autor, resulta un campo de investigación fascinante y factible sobre el que aún planean numerosas incógnitas.

A pesar de no ser novedoso, es importante mencionar la existencia de los SHA (Sistemas Hipermedia Adaptativos) por su influencia en este trabajo. Durante la década del 2000 se popularizaron en los ambientes de investigación académica los Sistemas Hipermedia Adaptativos aplicados a la enseñanza. Estos se basan en la idea de aproximar la enseñanza a la forma de aprender del alumno. Uno de sus pioneros es *Peter Brusilovsky*²¹ experto en aprendizaje adaptativo que en su artículo *Adaptive Hypermedia*²² de 1998 describe el estado del arte y los principios de funcionamiento de los mismos.

La pieza fundamental de estos sistemas es el motor de adaptación [14], [15]. Este programa identifica, a través de un cuestionario el tipo de aprendizaje que mejor se adapta al alumno. Es entonces cuando la inteligencia artificial del motor realiza las adaptaciones necesarias en el curso en línea con el fin de favorecer la adquisición de conocimientos y competencias. En algunas implementaciones, el motor analiza los progresos del alumno y reajusta el tipo de aprendizaje si detecta que no es adecuado. Las adaptaciones que realiza el sistema son básicamente de dos tipos: de interfaz y de contenido.

Las adaptaciones de la interfaz consisten en adaptar el curso a las características del dispositivo donde se visualiza. Dentro de este tipo tendríamos las adaptaciones a la resolución de la pantalla, las capacidades del navegador, el tipo de aparato... Son la más sencillas de implementar. También es posible modificar el sistema de navegación para que vayan mostrándose las nuevas lecciones a medida que el alumno supera las anteriores; por ejemplo habiéndolas visitado o superando una evaluación.

Las adaptaciones de contenido son las más complejas. Consisten en adecuar los contenidos a las características del alumno. Habrá alumnos que asimilen los contenidos educativos más rápidamente a través de la visualización de un vídeo, para otros es más efectivo leer una explicación en la que predominen los ejemplos o la teoría, etc. Otras posibilidades son ofrecer explicaciones adicionales que reflejen diferentes niveles de profundización en los contenidos, o que en función del tipo de alumno y sus conocimientos, este pueda escoger libremente el orden de acceso o se le orienta por el curso en función de sus progresos.

En la práctica, las adaptaciones de contenido son muy difíciles y complejas de implementar. No tanto por el motor de adaptación, sino por la gran cantidad de versiones que hay que hacer para cada contenido. Aunque no resulten viables en el terreno de la formación, sí se aplican en otros como el de la publicidad por internet, sistemas de navegación o búsquedas asistidas, diseño adaptable (*responsive design*), videojuegos... Por ejemplo, es frecuente que la publicidad que aparece en las páginas web esté relacionada con las búsquedas o páginas que hemos visitado; cuando no sea directamente de la tienda en línea que frecuente el visitante.

De todas las tecnologías que aparecen en este capítulo, esta última es, sin duda, la que más ha influido en los planteamientos del presente trabajo. Algunas de las ideas se han reflejado en la plataforma de publicación de los cursos desarrollada como la existencia de una explicación base de los temas tratados que se complementa con ventanas donde se amplían conceptos. El sistema de navegación permite a los usuarios seguir el orden sugerido o navegar libremente por el curso. En los ejercicios se ocultan las resoluciones para que sea el alumno el que decida voluntariamente cuando desea consultarlas. Y hay explicaciones interactivas y animaciones que ofrecen otras maneras de aprender.

Por otro lado, se ha dado una vuelta de tuerca al problema de la adaptación de los contenidos. Si no es factible adecuar el contenido al alumno, ¿por qué no darle las herramientas para que lo haga el mismo? Así, podrá ajustarlos a su forma de estudiar. La idea es que, sin que se pierdan los contenidos originales, el usuario pueda crear sus propias capas de

²¹ <http://www.pitt.edu/~peterb/>
https://en.wikipedia.org/wiki/Peter_Brusilovsky

²² http://www.cs.odu.edu/~jbollen/spring03_IR/readings/brusilovsky2001.pdf

contenidos personalizados con el máximo de libertad posible. Es lo mismo que los alumnos de la era pre-internet hacían cuando estudiaban: tomar apuntes, hacer anotaciones sobre los libros de texto, etc. pero trasladado a las TIC.

4 El medio

A comienzos del siglo XXI, Internet, y en concreto el WWW, se erigen como el medio de comunicación más importante si nos atenemos a su implantación y proyección de futuro. Es un medio masivo donde se ofrece a los usuarios todo tipo de servicios en los que puede participar como consumidor o creador. Esta democratización de los contenidos permite acceder a casi cualquier tipo de información imaginable y a la vez dificulta la visibilidad de los mismos por la gran cantidad de ellos existentes.

Internet tiene un alcance casi universal (al menos en el mundo desarrollado), por su disponibilidad temporal (todos los días y a todas horas), geográfico-tecnológica (desde cualquier lugar y a través de una amplia gama de dispositivos como ordenadores, teléfonos inteligentes, vehículos...) e idiomática (el soporte multilinguaje y las herramientas de traducción automática, cada vez más perfeccionadas, facilitan el acceso y comprensión de los contenidos). Y además, las tecnologías existentes permiten crear ricas y variadas aplicaciones donde el límite es la imaginación de los desarrolladores.

Como herramienta de formación, Internet es un hervidero de propuestas. Se puede encontrar desde simples vídeos explicativos de las cosas más simples y banales a campus virtuales y cursos con temarios propios de materias universitarias. La oferta es muy variada.

Este capítulo se va a centrar en qué es Internet, como funciona y qué consideraciones hay que tener en cuenta al desarrollar un proyecto en este medio.

4.1 Internet

Internet es una red de ordenadores descentralizada que engloba redes públicas, domésticas, gubernamentales, educativas, comerciales, etc. que se comunican entre sí mediante un conjunto de protocolos llamado "*Protocol Internet Suite*" donde se detallan todos los aspectos necesarios para gestionar la transmisión de la información. [16]

La comunicación se realiza mediante un proceso llamado conmutación de paquetes en el que se divide la información en bloques de datos que llevan incorporada la dirección de destino y la ruta a seguir.

El Protocolo de Internet (IP) funciona sobre capas o *layers*, cada una de las cuales es responsable de un aspecto de la comunicación. Estas van desde la comunicación entre dispositivos físicos de una red (tarjetas de red ethernet, Wi-Fi, switches, routers...), hasta la especificación de los servicios a los usuarios finales. Son:

- Capa de enlace de datos (*link layer*)
- Capa de red (*Internet layer*)
- Capa de transporte (*transport layer*)
- Capa de la aplicación (*application layer*)

La capa de enlace se encarga de la comunicación a nivel físico entre los dispositivos o nodos de una misma red. Controla cómo se generan los paquetes de información y se envían de un punto a otro. Ejemplos de implementación de esta capa son Ethernet o Wi-Fi.

La capa de red es la responsable de interconectar los nodos de la red para enviar los paquetes de información desde el punto de origen al de destino. Este proceso se conoce como enrutamiento. Para ello es necesario:

- Identificar los dispositivos conectados a la red (puntos) de forma unívoca mediante direcciones IP²³.
- Enviar los paquetes (*packet routing*) desde el origen hasta el destino guiándolos al siguiente nodo de la red más próximo al nodo final.

La capa de transporte se encarga de garantizar la transmisión de datos de una aplicación, entre dos máquinas de la red. Para ello, se añade a los paquetes de datos las direcciones IP de los dispositivos de origen y destino, y el puerto²⁴ donde se especifica la aplicación que ha de gestionar la información recibida. Los protocolos más importantes de esta capa son TCP y UDP que se distinguen por la fiabilidad y el orden en que se entregan los paquetes.

TCP (*Transfer Control Protocol*) es el protocolo sobre el que funciona la WWW y se basa en los siguientes principios:

- Orientado a la conexión.
- Transferencia fiable:
 - Los paquetes se reciben en orden.
 - Los paquetes dañados o perdidos se vuelven a enviar.
 - Los datos se entregan libres de errores y sin duplicados.
- Se pueden definir parámetros de calidad de servicio.
- El destino soporta la utilización simultánea de más de un servicio de red (aplicación) mediante el uso de puertos.

El protocolo UDP (*User Datagram Protocol*) se utiliza cuando interesa primar la velocidad de transmisión de datos sobre la fiabilidad. No tiene control sobre el orden de transmisión de los datos ni sobre si han llegado todos los paquetes a su destino.

Finalmente, en la capa de la aplicación se definen los protocolos de comunicación de las aplicaciones y servicios de Internet que utilizan los usuarios u otras máquinas. Entre los muchos existentes destacan:

- Documentos: HTTP, HTTPS, Gopher.
- Acceso a archivos: FTP, TFTP.
- Servicios de red: DNS, DHCP.
- Correo: SMTP, POP, IMAP
- Acceso remoto a otros dispositivos: SSH, Telnet
- Telefonía sobre red: VoIP
- Intercambio de archivos: torrent, P2P, eDonkey.
- Chats: IRC.
- Televisión: IPTV.
- Juegos en línea.
- Etcétera

Internet es pues, la infraestructura de maquinaria y programas que ofrece conectividad a los ordenadores que la forman.

²³ La dirección IP es un identificador único de red. Según la versión IPv4 RFC791 [152] es un número binario de 32 bits. Ante el agotamiento de las combinaciones disponibles (4000 millones), se ha sustituido por la IPv6 RFC2460 [153] que usa un número de 128 bits.

En IPv4, las IP tienen la forma a.b.c.d donde a, b, c y d toman valores entre 0 y 255.

²⁴ "Un puerto de red es una interfaz utilizada para comunicarse con un programa a través de una red. Los puertos de red suelen estar numerados [...]; la implementación del protocolo en el destino utilizará este número para decidir a qué programa entregar los datos recibidos." [74]

4.2 World Wide Web (WWW)

WWW es un servicio para compartir documentos relacionados mediante enlaces de hipertexto según la arquitectura cliente-servidor. Es común confundir el WWW con Internet en el lenguaje común, pero son cosas diferentes.

El WWW se basa principalmente en el uso de tres tecnologías:

- *Uniform Resource Identifier* (URI) o *Uniform Resource Locator* (URL), que dota de un identificador único a cada documento o elemento de un sitio web.
- *Hyper Text Markup Language* (HTML), que es el lenguaje de marcas que se emplea para codificar los documentos. Para aumentar la riqueza e interactividad de estos, se emplean también las hojas de estilo (*Cascading Style Sheets*, CSS) y los *scripts* (programas que añaden interactividad).
- El protocolo HTTP que define cómo deben realizarse las peticiones de documentos del cliente al servidor y cómo ha de ser la respuesta de este.

Para navegar por la red se utilizan programas especializados llamados navegadores, exploradores o *browsers* que gestionan las peticiones de documentos al servidor web.

Cuando un usuario introduce una dirección URI en el programa, se desencadena un complejo proceso que acaba por mostrar la página web que vemos en la pantalla.

En primer lugar, el navegador crea una petición HTTP que envía al servidor. Este responde, enviándole el documento HTML si está disponible y un mensaje de error en caso contrario. El navegador analiza el código y solicita al servidor el resto de elementos necesarios para generar lo que vemos en pantalla (CSS, imágenes, *scripts*, vídeo...). Finalmente el navegador genera la representación gráfica de la página en un proceso llamado **renderizado**.

```
<!DOCTYPE html>
<html>
<head>
<title>Activitat 1</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<meta http-equiv="content-type" content="application/xhtml+xml; charset=UTF-8">
<meta http-equiv="content-style-type" content="text/css">
<meta http-equiv="expires" content="0">
<link href="css/activitat1.css" rel="stylesheet" type="text/css">
</head>
<body>
<header>
<h1>Activitat 1</h1>
</header>
<main>
<form>
<section id="client">
<fieldset>
<legend>Dades del client</legend>
<label>CIF <input type="text" size="12" name="cif" pattern="[XY\d]\d{7}[A-Z]$" /></label><br/>
<label>
Nom <input type="text" size="16" name="nom" list="cliente" />
<datalist id="cliente">
<option>Alpha SL</option>
<option>Beta SL</option>
<option>Gamma SL</option>
<option>Epsilon SL</option>
</datalist>
</label>
</fieldset>
</section>
<section id="producte">
<fieldset>
<legend>Dades del producte</legend>
<label>Referència <input type="text" size="5" name="ref" pattern="[A-Z]-\d{4} $" /></label><br/>
<label>Producte
<select name="producte">
```

```

<optgroup label="Cadires">
<option value="1">Cadira de rodes</option>
<option value="2">Cadira</option>
</optgroup>
<optgroup label="Taula">
<option value="3">Taula rodona</option>
<option value="4">Taula quadrada</option>
</optgroup>
</select><br/>
Forma de pagament
<label><input type="radio" name="pagament" value="efectiu" checked="checked" /> Efectiu</label>
<label><input type="radio" name="pagament" value="targeta" /> Targeta</label>
</label>
</fieldset>
</section>
<section id="botons">
<input type="submit" value="enviar" size="35" /><br/>
<input type="reset" value="reiniciar" size="35" />
</section>
</form>
</main>
<footer>
<ul>
<li>Clients d'avui</li>
<li><a href="">Avis legal</a></li>
<li><a href="">Contacte</a></li>
</ul>
</footer>
</body>
</html>

```

Código 1 Ejemplo de página HTML

Activitat 1

Ilustración 3 Renderizado del código HTML anterior

El WWW está regulado por la *World Wide Web Consortium* o W3C que es la responsable de la elaboración de los estándares, recomendaciones y normas. Trabaja en colaboración con otras instituciones o grupos de trabajo que se encargan de desarrollar proyectos complementarios como javascript (ECMA), HTML5 (*Web Hypertext Application Technology Working Group* - WHATWG), la definición de URL (*Internet Engineering Task Force*-IETF)...

4.2.1 Arquitectura cliente servidor

Es la implementación de un sistema distribuido²⁵ basado en la existencia de dos tipos de nodos en la red: clientes y servidores.

²⁵ Un sistema distribuido consiste en un conjunto de máquinas, cada una de ellas especializada en realizar una tarea, que trabajan juntas para procesar e intercambiar información. De esta manera el trabajo se reparte entre diferentes ordenadores.

El cliente es el elemento activo de la red; un consumidor de información. Realiza las solicitudes de documentación al servidor, espera la respuesta y luego muestra el resultado con ayuda de un software especializado (*browser*).

El servidor, por el contrario, es el elemento pasivo de la red que sólo actúa cuando recibe una solicitud del cliente. Es un proveedor de información. Un software especializado instalado en él (llamado servicio en entornos Windows o *daemon* en entornos Linux-UNIX) se mantiene a la escucha, esperando a recibir peticiones para procesarlas y emitir la respuesta.

La implementación puede ser de dos capas o multicapa.

- La implementación de dos capas se produce cuando en el lado del servidor sólo interviene una máquina.

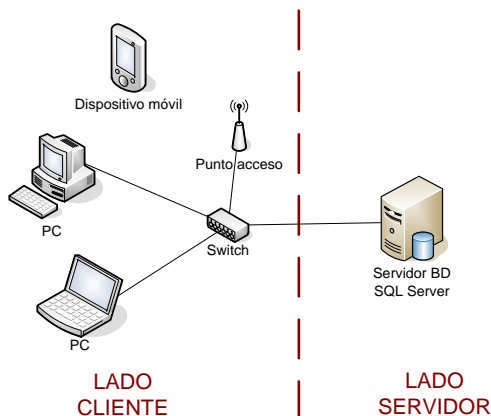


Ilustración 4 Arquitectura cliente-servidor de dos capas

- En la multicapa el servidor actúa a su vez como cliente de otros servidores. Cada servidor puede especializarse en una tarea determinada del proceso (archivos, bases de datos, mensajería...), se aumenta así la velocidad de respuesta y se mejora el mantenimiento y la seguridad.

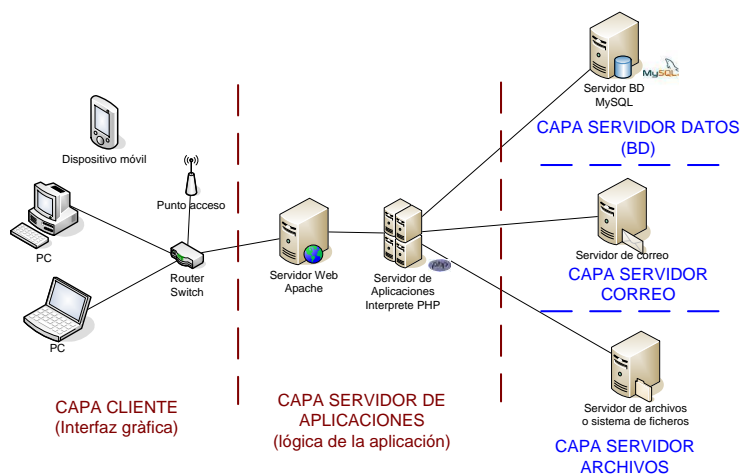


Ilustración 5 Arquitectura cliente-servidor multicapa

4.2.2 El protocolo HTTP

Las páginas web son documentos hipermedia²⁶ e hipertexto²⁷ alojados en servidores remotos que se consultan desde el dispositivo del cliente. La sintaxis de los mensajes está especificada en el protocolo HTTP en el caso del *World Wide Web*²⁸ (WWW) o *Web*.

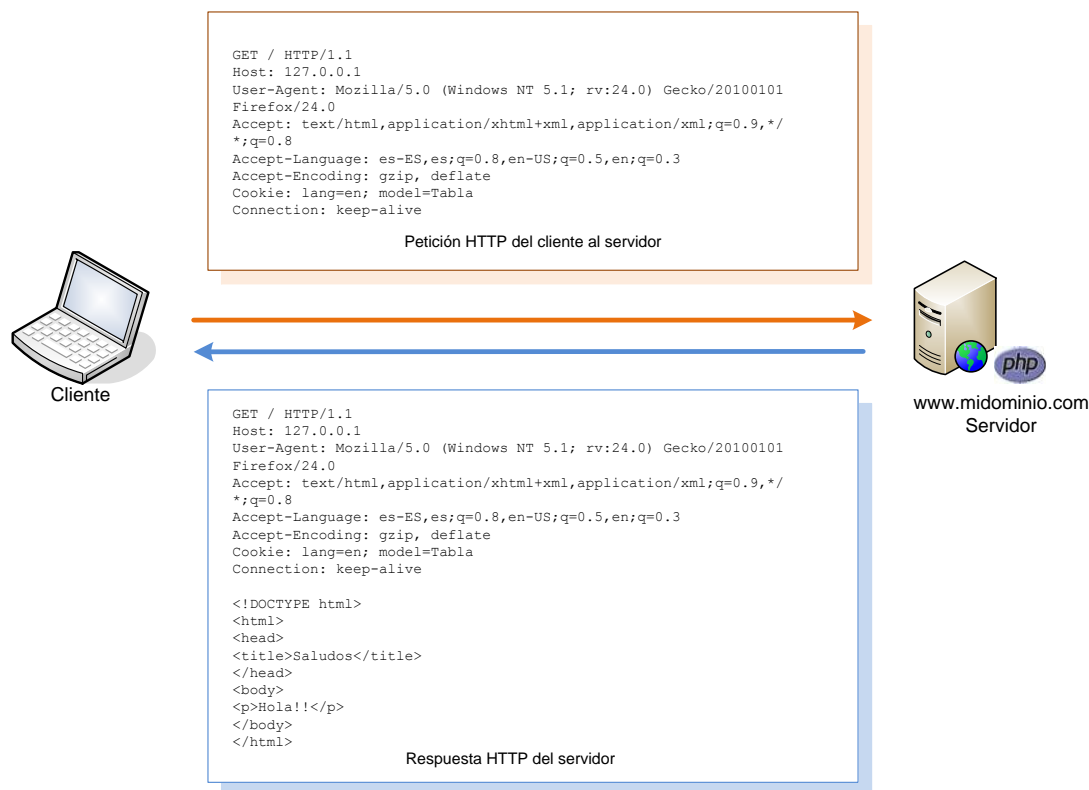


Ilustración 6 Esquema de la comunicación cliente-servidor

El protocolo HTTP consiste en un conjunto de instrucciones en texto plano²⁹ conocidas como transacciones. Se utilizan para enviar y recibir las páginas web y el resto de archivos entre el cliente y el servidor. Una de las características más significativas es que es un protocolo sin estado. Por lo que el servidor, una vez envía la respuesta, descarta toda información sobre la petición que la generó.

Una transacción HTTP está formada por el encabezado, una línea en blanco y los datos. El encabezado consta de la línea de petición si la comunicación va del cliente hacia el servidor, o en caso contrario, de la línea de estado de la respuesta. Le siguen las cabeceras *header* adicionales en las que se incluye información extra. Los datos sólo se incluyen si hay información que transmitir; por ejemplo, cuando se envía un formulario al servidor o el servidor envía el código HTML de la página solicitada. La norma completa puede consultarse en la RFC 2616 [17].

²⁶ Documento que combina contenidos en diferentes formatos como texto, vídeo, audio, etc. y permite al usuario interactuar con ellos. [146]

²⁷ "El hipertexto es un sistema de organización de la información basado en la posibilidad de moverse por dentro de un texto y hacia textos diferentes por medio de palabras clave." [147]

²⁸ "The *World Wide Web* (WWW, or simply *Web*) is an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI)." [171]

²⁹ Conjunto de texto sin formato ni estilo formado sólo por caracteres y que es legible y comprensible para los seres humanos.

A modo de ejemplo, se muestran las transacciones HTTP para la dirección web <http://127.0.0.1> en un servidor de desarrollo XAMP³⁰ en modo local³¹.

```
GET / HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:24.0) Gecko/20100101 Firefox/24.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: lang=en; model=Tabla
Connection: keep-alive
```

Código 2 Petición HTTP cliente-servidor

```
HTTP/1.1 302 Found
Date: Sun, 10 Nov 2013 11:14:30 GMT
Server: Apache/2.4.4 (Win32) OpenSSL/0.9.8y PHP/5.4.19
X-Powered-By: PHP/5.4.19
Location: http://127.0.0.1/xampp/
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<html>
...
[código HTML de la página]
...
</html>
```

Código 3 Respuesta HTTP servidor-cliente

En el ejemplo anterior se muestra un problema inherente al uso del protocolo HTTP: el de la seguridad. Como se utiliza texto plano para escribir los mensajes, resulta susceptible de ser interceptado por algún software especializado, como pueden ser los analizadores de paquetes o *sniffers*³². Esta falta de seguridad no es grave si los contenidos mostrados son de dominio público. El inconveniente aparece cuando lo que se transfiere es información sensible o de carácter privado. Para estos casos es conveniente utilizar el protocolo HTTPS³³ que implementa un sistema de cifrado de comunicaciones.

4.3 Accesibilidad

Uno de los problemas más acuciantes de la WWW lo constituyen las barreras de accesibilidad a los contenidos. Aunque pueda parecer lo contrario, este no es un problema exclusivo de las personas con discapacidad, sino que afecta a todos los usuarios en mayor o menor medida.

Se trata, en definitiva, de poner a disposición de todo el mundo la web, y sus contenidos, rompiendo las barreras que limitan su acceso y uso: idioma, economía, nivel educativo, localización geográfica, tecnología de acceso...

Las barreras se pueden clasificar en aquellas que afectan a:

³⁰ Entorno de desarrollo para aplicaciones de servidor. Desarrollado originariamente para entornos Windows con Apache, PHP y MySQL aunque actualmente soporta otros sistemas operativos, lenguajes de servidor y servicios como correo, ftp... <http://www.apachefriends.org/index.html>

³¹ Software de servidor que se ejecuta la misma máquina que el cliente.

³² Programa capaz de capturar cualquier trama o paquete de información que circula por la red. [123]

³³ Protocolo seguro de transferencia de hipertexto.

- Los dispositivos
- Los navegadores
- Las personas con discapacidad

4.3.1 Barreras debidas a las características de los dispositivos de los usuarios

Son aquellas que se producen cuando se desarrolla una web pensando básicamente en un determinado ecosistema; ya sean dispositivos móviles o de escritorio. Cada sistema tiene sus particularidades y conviene darle un tratamiento específico.

En el caso de dispositivos móviles, sus particularidades están asociadas al tamaño y a la resolución de su pantalla. Ésta, al ser de un tamaño reducido, no posibilita la gestión del espacio entre los contenidos de la misma forma que en la pantalla de un ordenador de escritorio con alta resolución. Claro que los móviles disponen de función de zoom, pero resulta incómodo para los usuarios. La solución es adaptar los contenidos al medio. Esto es, suprimir y/o reducir contenidos selectivamente como, por ejemplo, la publicidad; cambiar la distribución de los elementos, etc.

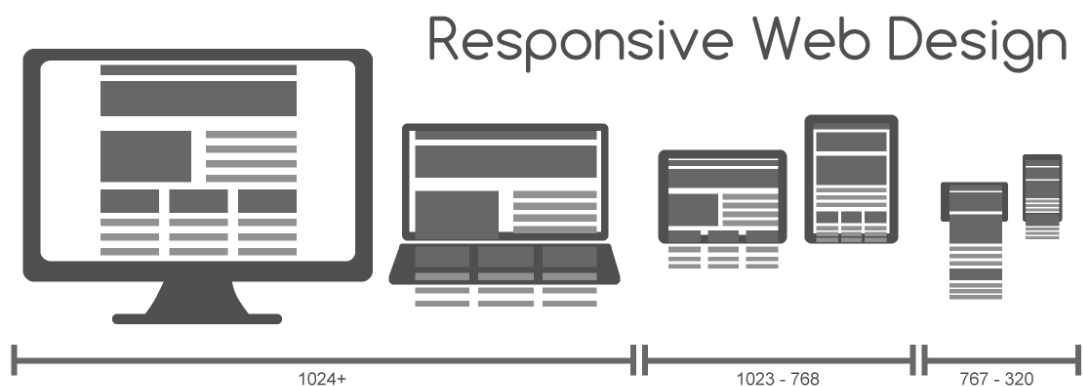


Ilustración 7 Diseño adaptable a las características del dispositivo³⁴

Estas particularidades no afectan sólo a los contenidos, sino también a las características de la interfaz del usuario³⁵ puesto que un dispositivo táctil no funciona exactamente igual que el ratón de una máquina de escritorio o un teclado.

No es una barrera insalvable, con CSS3 ya es posible definir la distribución en función de las características del dispositivo. Y para los navegadores más antiguos, se puede recurrir a javascript.

4.3.2 Barreras debidas a las características de los navegadores

Las barreras de accesibilidad en los navegadores están relacionadas con los dispositivos, pero afectan sobre todo a la capacidad de éstos para interpretar HTML, CSS y javascript. Entre las causas están que no existe un único explorador en el mercado, sino varios; a que los estándares están en constante evolución y a que la implementación de los cambios no es uniforme. De esta manera, el desarrollador web se enfrenta a un grave problema de compatibilidad que consume mucho tiempo y recursos. Por suerte, los desarrolladores de navegadores están convergiendo en la aplicación de los estándares y cada vez es más fácil ofrecer una experiencia de uso similar a todos los usuarios de forma sencilla.

³⁴ Fuente: <http://www.grapefruit.ro/ideas/what-is-responsive-design/> [140]

³⁵ "La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo." [149]

Los problemas más frecuentes son:

- Características de HTML no implementadas como sucede con algunas etiquetas y atributos de HTML5; por ejemplo, algunos de los nuevos controles para formularios.
- Propiedades de CSS no implementadas. Algunas características de CSS3 como párrafos multicolumna, imágenes en los bordes...
- Métodos y propiedades de javascript no implementados. Ocurre tanto con ECMA-script, como con DOM o con las API de HTML5.

La mayoría de estos problemas se pueden solventar mediante complicadas implementaciones de javascript e incluso existen *frameworks* como jQuery nacidos específicamente con este propósito.

Se puede recurrir también a un diseño escalable de la aplicación. Se crea una versión básica que se vea dignamente aún en las peores configuraciones y se van incrementando las prestaciones de la página a medida que aumentan las capacidades del navegador.

Conviene en todos los casos, definir al inicio del proyecto cuáles serán las versiones y navegadores en las que se centraran los esfuerzos de desarrollo y cuáles quedan fuera del proyecto. Una buena herramienta, es una matriz de compatibilidad donde se reflejarán las decisiones tomadas.

Navegador	Versió	Windows	OS X	Linux	iOS	Android
Internet Explorer	6	NO	N/A	N/A	N/A	N/A
	7,8	NO	N/A	N/A	N/A	N/A
	9,10	SI	N/A	N/A	N/A	N/A
	11	SI	N/A	N/A	N/A	N/A
Firefox	31 (±2)	SI	SI	SI	N/A	SI
Chrome	35 (±2)	SI	SI	SI	SI	SI
Safari	6	N/A	SI	N/A	SI	N/A
Opera	22 (±2)	SI	SI	SI	SI	SI
Android Browser	--	N/A	N/A	N/A	N/A	SI

Ilustración 8 Matriz de compatibilidad de navegadores web³⁶

4.3.3 Barreras para las personas

Finalmente, están las barreras a la accesibilidad que impiden usar la web con normalidad y afectan básicamente a las personas con discapacidad y en menor medida al resto.

El reto está en conseguir que cualquier persona independientemente de sus circunstancias pueda: navegar por la red, interactuar con las aplicaciones, comprender los contenidos, percibir las páginas con alguno de sus sentidos, disfrutar de los contenidos multimedia, y aportar contenidos a la web.

Resolver esta cuestión beneficia tanto a las personas con alguna discapacidad (sea visual, auditiva, física, cognitiva, neurológica o de habla), como a personas que han visto reducidas sus capacidades por la edad, enfermedad o cualquier otra circunstancia de forma permanente o temporal. También supone una mejora para las personas sin discapacidad, puesto que cuando se diseña una *web* se contemplan las características de la conexión (velocidad y ancho de banda), el dispositivo utilizado (móvil, tableta...) y el *software* (navegadores antiguos). Otros beneficiados con el la eliminación de barreras para las personas, son aquellas con dificultades de comprensión del lenguaje, ya sea por tener un nivel educativo bajo o no ser lo suficientemente competentes en el idioma (inmigrantes, turistas, niños...). Por último, las organizaciones también están interesadas en proporcionar la máxima accesibilidad a las personas puesto que lleva

³⁶ Página 33 [93]

consigo implicaciones en temas de optimización de búsquedas (*Search Engine Optimization*, SEO), fidelización de clientes, imagen (responsabilidad social corporativa) y aspectos legales.

Eliminar estos obstáculos implica invertir tiempo y dinero, supone aumentar la complejidad de las web y resulta tedioso para los desarrolladores; pero también aporta beneficios que compensan sobradamente las dificultades.

La WAI (*Web Accessibility Initiative*)³⁷ es una organización englobada dentro del W3C que se encarga de promover la accesibilidad en Internet. Emite recomendaciones y normas para hacer la web accesible a todo el mundo. En su sitio web se explican cuatro clases de beneficios que aporta la accesibilidad a las organizaciones. [18]

- Factores sociales
- Factores técnicos
- Factores económicos
- Factores legales

Los factores sociales se refieren a todas las personas que se benefician por aplicar las políticas y técnicas de accesibilidad a las webs. En resumen, suponen ampliar la base de usuarios potenciales, mejorar la usabilidad y navegación, aplicar políticas de responsabilidad social corporativa y fomentar la igualdad de oportunidades para acceder y disfrutar de la información.

Los factores técnicos resultan de la aplicación de estándares web y recomendaciones de accesibilidad. En primer lugar, obtenemos una reducción en los tiempos de desarrollo y mantenimiento del web gracias al uso de técnicas de diseño escalable que combinadas con CSS y otras técnicas hacen más sencillo adaptar la web a las características del dispositivo y/o del usuario. En segundo lugar, se produce una disminución de la carga de trabajo del servidor y el ancho de banda necesario, puesto que se reduce la navegación ociosa y el tamaño de los archivos (por ejemplo, se sustituyen imágenes por texto). Por último, al ser la transversalidad tecnológica uno de los principios de la WAI, es más fácil adaptar la web a las futuras tecnologías.

Los factores financieros son resultado de aplicar lo anteriormente expuesto. Así, se consigue un incremento potencial de beneficios y objetivos aumentando el público objetivo, una mejora del ranking en los buscadores ya que las técnicas de accesibilidad implican utilizar a su vez técnicas de SEO y del grado de satisfacción de los usuarios. Otros beneficios asociados son el ahorro en costes de personal, mantenimiento y desarrollo y con la reducción de costes asociados a la emisión de facturas electrónicas, en las traducciones a otros idiomas utilizando un lenguaje simplificado...

Los factores legales implican el cumplimiento de las leyes y normativas vigentes, así como las sanciones que conllevan su infracción.

Hacer accesible el WWW es un trabajo conjunto en el que intervienen los usuarios, desarrolladores web, de navegadores, de herramientas de desarrollo, de validación y de tecnologías de soporte. Cada uno es responsable de una parte del proceso.

³⁷ <http://www.w3.org/WAI/>

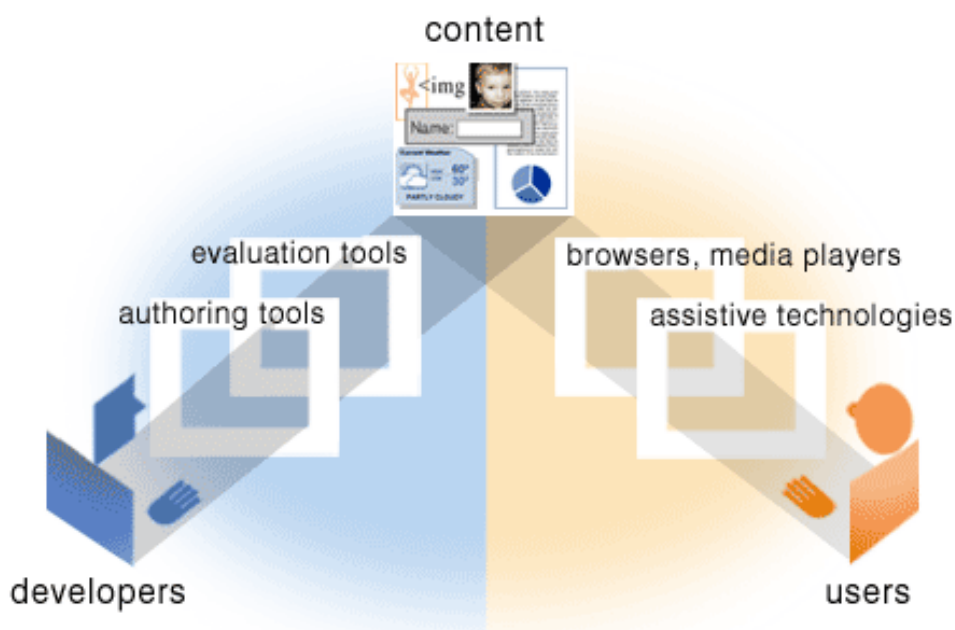


Ilustración 9 Actores que participan de la accesibilidad³⁸

Algunas medidas implican la participación activa del usuario como el uso de tecnologías de soporte. Estas consisten en un conjunto de dispositivos y programas que mejoran la accesibilidad. A modo de ejemplo tenemos programas que leen el texto en pantalla, visualización en alto contraste, control del tamaño de la letra, lupa, dispositivos de entrada alternativos al ratón como teclados especiales u órdenes por voz...

Otras van asociadas al diseño web, como la aplicación de buenas prácticas de diseño (navegación clara, técnica de las migas de pan, contraste de colores...) e implementación de la recomendaciones.

Con el fin de hacer accesible una web, el WAI ha emitido una serie de guías que cubren diferentes ámbitos del diseño y desarrollo web.

- Contenido. WCAG. *Web Content Accessibility Guidelines*
- Implementación de navegadores y reproductores multimedia. UAAG. *User Agent Accessibility Guidelines*
- Implementación de herramientas para crear páginas web. ATAG. *Authoring Tool Accessibility Guidelines*
- Implementación de aplicaciones web (AJAX, javascript, DOM ...) ARIA. *Accessible Rich Internet Applications*
- Definición de interfaces independientes del dispositivo y/o la tecnología empleada por el usuario para interactuar. IndieUI. *Independiente Interfaz*
- Formato para expresar los resultados de la evaluación. EARL. *Evaluation and Report Language*

Las más importantes para el desarrollador web son la WCAG³⁹ que es una guía para el desarrollo de contenidos y la ARIA⁴⁰ que ayuda a las tecnologías de soporte y a los navegadores informando de cómo se estructura la página web y qué papel juegan los elementos interactivos.

³⁸ Fuente: WAI. <http://www.w3.org/WAI/intro/components.php>

³⁹ <http://www.w3.org/TR/WCAG20/> [53]

⁴⁰ <http://www.w3.org/TR/wai-aria/> [174]

4.4 Paradigmas de programación aplicables

Para conseguir una buena optimización de recursos cuando se diseña una página es necesario aplicar una metodología durante todo el proceso de diseño y desarrollo. Así se evitan o se reducen al máximo posible los cambios durante la implementación.

Además de los temas de accesibilidad y diseño adaptable que ya se han tratado anteriormente, hay otras consideraciones que merecen ser explicadas con un poco de detalle.

4.4.1 Promoción de la web

Tan importante como hacer una web con un contenido interesante, una buena navegación o que sea accesible, es conseguir que tenga visitas. Para que esto ocurra, es fundamental la promoción que consiste básicamente en mejorar su posición en los buscadores como Google, Bing, Yahoo, Ask, DuckDuckGo...

Los buscadores son herramientas que ayudan a los usuarios a encontrar información útil en la web. Complejos algoritmos consultan inmensas base de datos para devolver, en un breve espacio de tiempo, una lista de resultados **relevantes** ordenados por **popularidad**. Previamente, la información se ha recopilado mediante el uso de robots (*web robots*, *web wanderers*, *crawlers* o *spiders*) que recorren continuamente la web siguiendo los enlaces de las páginas e indexando sus contenidos.

El concepto de información relevante es importante, puesto que supone que los resultados han de aproximarse lo más posible a lo que busca el usuario. Por ejemplo, si buscan restaurantes de comida japonesa, la lógica indica que estén próximos a la ubicación física del usuario.

Los algoritmos de indexación utilizan multitud de variables llamadas factores de ranking para establecer la clasificación final. Aunque son secretos, existen guías que dan pautas sobre cómo mejorar el ranking. [19], [20], [21]

Para que todo el proceso tenga sentido, hay que entender cómo las personas utilizan los buscadores. En general, las búsquedas se pueden clasificar en tres tipos [21] :

- Quiero hacer una cosa
- Quiero información sobre algo
- Quiero ir a un sitio web

Una vez obtenida la lista de resultados, el usuario mira entre los enlaces obtenidos aquellos que se ajusten mejor a lo que busca. Después de probar uno, o varios, enlaces, si no encuentra lo esperado refinará la búsqueda o iniciará una nueva. Determinar lo que buscan los usuarios en nuestra web, pues, resulta fundamental para conseguir buenos resultados.

El SEO (*Search Engine Optimization*) es un conjunto de técnicas que tienen por objetivo mejorar la visibilidad y el posicionamiento orgánico⁴¹ de una web en los motores de búsqueda (*search engines*). En general, son reglas que apelan al sentido común: las hay que se refieren al contenido, la navegación, la estructura de la web, facilitar el trabajo del robot y la reputación de la publicación. Si la web es accesible, muchas de estas técnicas estarán implementadas de facto.

⁴¹ El posicionamiento orgánico es aquel que depende únicamente del motor de búsqueda, sin que sea alterado por el pago de cantidades, publicidad o patrocinio.

4.4.2 Diseño no intrusivo

Este paradigma de la programación separa el contenido, la presentación y la funcionalidad (aplicaciones) de las páginas web. De esta manera, se obtiene una estructura de código limpia donde cada parte se puede implementar de forma aislada.

Consta de tres elementos:

- El archivo HTML, en el que se incluye la información y la estructura semántica del documento (apartados, secciones, anuncios...)
- El archivo CSS donde se define cómo será la presentación del documento (tipografía, colores, distribución y medidas de los elementos...)
- El archivo javascript donde se implementa toda la funcionalidad de las aplicaciones definidas en la página web.

El documento HTML así construido está limpio de implementaciones de CSS y javascript, ya que éstos se encuentran en archivos separados. Las etiquetas HTML están limpias de controladores de eventos de javascript y de atributos de estilo propios de CSS difíciles de mantener. El documento resultante es más fácil de tratar por los robots de búsqueda y por las tecnologías de soporte. Se puede concluir, que utilizar este paradigma redundará en una mejora de la accesibilidad y el SEO. Los cambios en el contenido y la estructura implicarán necesariamente algunos cambios en el CSS y el Javascript, pero al revés no siempre será necesario.

Modificar el aspecto del documento, al estar centralizado en unos pocos archivos, o incluso en uno solo, resulta más fácil y práctico. La funcionalidad también sale beneficiada puesto que el producto resultante es más robusto, fiable y se desarrolla más rápidamente. Las tendencias actuales de programación en javascript hablan de diseño no obstructivo (*Unobtrusive JavaScript*) como un código robusto a los errores (verifica la existencia de funciones y HTML con los que interactúa, soluciona problemas de compatibilidad entre navegadores, en caso de error no provoca la interrupción de la ejecución de la página ...), modular (el código se encapsula en unidades que son independientes del resto de código utilizado en la página web) y que no ha de deteriorar el acceso a los contenidos aunque falle o esté desactivado (estos se han de ver en el peor de los casos; no puede quedar contenido oculto por culpa de un fallo en la ejecución). [22], [23], [24]

El código javascript resultante es más complejo. Hay que definir módulos autónomos donde se encapsula el código y controladores de eventos que gestionan la respuesta del programa ante las acciones del usuario como pulsar botones, pero las ventajas superan los inconvenientes.

4.4.3 MVC

El Modelo-Vista-Controlador o MVC es un patrón de arquitectura de *software* que se utiliza para programar interfaces de usuario (UI). Se basa en separar la lógica de negocio (cómo se gestiona internamente la información), de la presentación e interacción con el usuario.

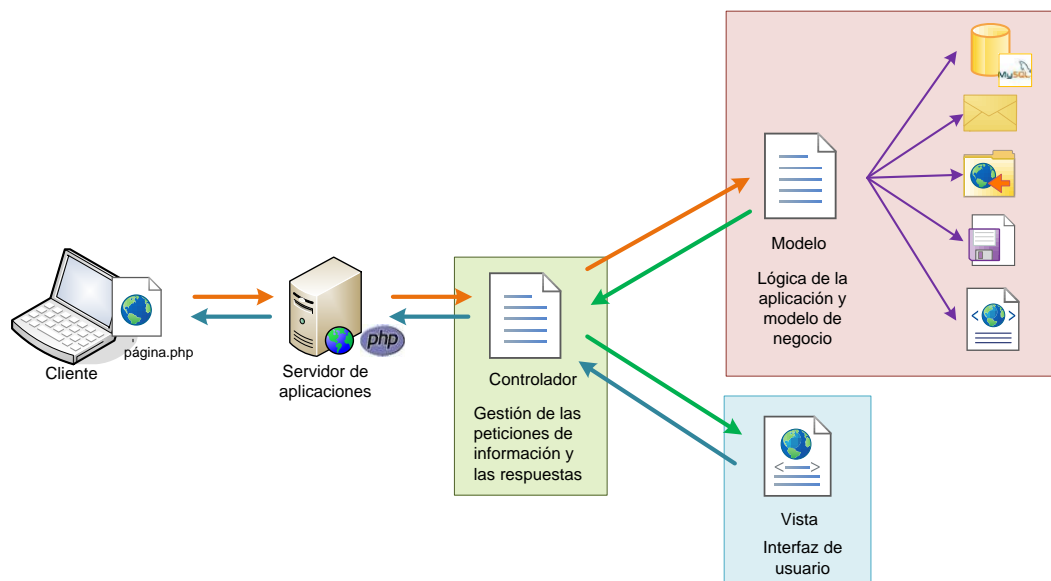


Ilustración 10 Esquema de funcionamiento de la arquitectura MVC

Utilizar MVC reporta las siguientes ventajas: el código es más modular, reutilizable y resulta más cómodo y fácil de mantener. Las modificaciones son más sencillas, pues solo afectan a los módulos requeridos. El trabajo se puede dividir en porciones que se pueden desarrollar de forma independiente y especializada. Se evita mezclar la lógica de negocio y la presentación y así, podemos definir perfiles de trabajo especializados como diseñadores y programadores de bases de datos, programadores de lógica de negocio en el entorno servidor, programadores de aplicaciones en el entorno cliente, maquetadores de HTML y CSS, diseñadores gráficos... que pueden trabajar de forma autónoma.

Con esta finalidad, el código se divide en tres partes (controlador, modelo y vista), cada una de ellas encargada de realizar una tarea específica.

El modelo representa la lógica de negocio. Se encarga de realizar los cálculos, acceder a las bases de datos y manipularlos, acceder a ficheros, enviar correos, etcétera. Recibe los datos del controlador y le devuelve la información necesaria para generar la vista en bruto, sin manipular.

La vista es la encargada de generar y actualizar la UI que verá el usuario en la pantalla de su navegador. En aplicaciones web, la vista puede generar muchos tipos de formatos además de HTML, por ejemplo imágenes, PDF, JSON, XML, ZIP... La implementación varía según el tipo de aplicación. En el caso de clientes ligeros (*thin client*)⁴² la generación de la vista se hace exclusivamente en el servidor que crea una página HTML que se envían al cliente; aunque también es posible utilizar otros tipo de archivo como PDF o imágenes. En los clientes pesados, cliente y servidor se intercambian principalmente datos en formato XML, JSON u otros mediante mediante AJAX. La interfaz se genera con javascript en el cliente. De esta manera se aumenta la velocidad carga, pues sólo se envían unos pocos datos en lugar de todo el código de la página.

El controlador realiza una función de intermediario entre el usuario, el modelo y la vista. Recoge las peticiones del cliente (ya sean solicitudes de información o alteración de los datos), las trata y se las envía al modelo para que realice las actualizaciones. Y cuando este termina, envía órdenes a la vista para que actualice la interfaz del usuario. En aplicaciones web, las peticiones llegan a través del nombre del archivo que se quiere visitar, de los datos enviados desde un formulario por GET o POST, desde de un enlace dinámico (QUERY_STRING), de archivos subidos, de información almacenada en *cookies* o de la sesión del servidor.

⁴² Un cliente pesado solicita al servidor los datos para procesarlos en la aplicación cliente que genera la presentación. Por el contrario, un cliente ligero realiza unas pocas operaciones y es el servidor el que realiza casi toda la carga de cálculo y genera la presentación.

En los clientes pesados, la implementación difiere un poco del ligero, pues la vista se reparte entre el cliente y el servidor. En el servidor, se generan los datos formateados que se envían navegador del cliente donde se reciben y se utilizan para modificar la página con javascript y DOM.

4.4.4 Programación orientada a objetos

La programación orientada a objetos (POO u OOP en inglés) es una forma de organizar el código o paradigma en que se agrupan dentro de una estructura llamada objeto todas las variables y funciones necesarias para modelizar una entidad del mundo real. Las variables se conocen como propiedades y las funciones como métodos.

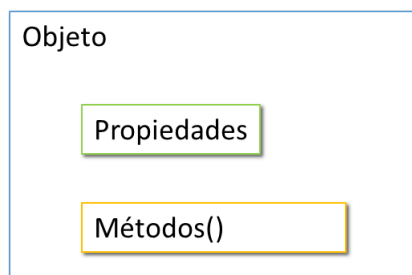


Ilustración 13 Estructura de un objeto en POO

Los objetos se construyen a partir de clases, que son la implementación en código fuente de su estructura. Cuando el programa crea una variable con el tipo de dato definido en una clase, utiliza ésta como modelo para crear un objeto. Por ello se dice que un objeto es una instancia de una clase. Aunque habitualmente se confunden ambos términos, los programas trabajan con objetos, no con clases. Haciendo un símil, podríamos pensar en la clase como un plano de una pieza y en los objetos como cada una de las piezas construidas siguiendo el plano.

En la programación orientada a objetos (POO) el código fuente está caracterizado por:

- Encapsulamiento
- Herencia.
- Polimorfismo

El encapsulamiento consiste en que cada clase incluye todas las propiedades y métodos necesarios para trabajar con un objeto. Esto permite que cualquier programador utilice una clase sin necesidad de conocer su implementación. Basta con saber con qué propiedades puede trabajar y cómo funcionan los métodos (parámetros, excepciones y valores devueltos).

```
class Rectangulo {
  // atributos - propiedades
  protected $a;
  protected $b;

  public function __set($name, $value){
    $this->$name=$value;
  }

  public function __get($name){
    return round($this->$name,4);
  }

  // constructor
  function __construct() {
    if(func_num_args()==2) {
      // amb argumentos
      $this->a=func_get_arg(0);$this->b=func_get_arg(1);
    }else {
      // sin argumentos
    }
  }
}
```

```

        $this->a=0.;$this->b=0.;
    }
}

// otros métodos
public function Area(){ return round($this->a*$this->b,4);}

public function Perimetro(){ return round(2.*$this->a+2.*$this->b,4);}

public function Diagonal(){ return round(sqrt($this->a*$this->a+$this->a*$this->b),4);}

}

```

Código 4 Ejemplo de implementación de una clase en PHP

En el ejemplo anterior, se ha implementado una clase para trabajar con rectángulos. Se han definido dos propiedades y seis métodos. En las propiedades se guardan los valores de las longitudes de los lados. Tres de los métodos sirven para hacer cálculos relacionados con la figura: diagonal, área y perímetro. Y los otros tres son el constructor que crea una instancia de la clase y la inicializa, *set* que da valores a los catetos y *get* que los devuelve.

La herencia permite crear clases a partir de otras existentes y modificarlas según nuestras necesidades. No es necesario reescribir el código completo; solo lo que sea preciso. Las propiedades y métodos de la clase padre estarán automáticamente disponibles, con excepciones, en las hijas. También se pueden incorporar nuevas variables y métodos. La herencia puede ser simple si cada clase hija sólo puede tener una clase padre como en Java, PHP, .net... o múltiple como en C++.

```

class Cuadrado extends Rectangulo{
    public function __construct() {
        if(func_num_args()===0){
            parent::__construct();
        }elseif (func_num_args()===1) {
            $this->a = func_get_arg(0);
        }else {
            $this->a = 0.;
        }
    }

    public function Area(){ return round($this->a*$this->a,4);}

    public function Perimetro(){ return round(4.*$this->a,4);}

    public function Diagonal(){ return round($this->a*sqrt(2),4);}

}

```

Código 5 Ejemplo de clase que hereda de la clase rectángulo

En el ejemplo anterior, se crea la clase cuadrado que hereda de rectángulo. En esta clase no se definen las propiedades ni los métodos *get* ni *set* que se heredan. El resto de métodos se reescriben con unas implementaciones más adecuadas.

El polimorfismo consiste en que dentro de una clase, varios métodos compartan el mismo nombre pero tengan implementaciones diferentes. En la mayoría de lenguajes esto se consigue con alguna de las siguientes técnicas:

- La sobrecarga se produce cuando dos o más métodos que comparten el mismo identificador se distinguen por el número y/o el tipo de los parámetros que se pasan a la función en la llamada. En algunos lenguajes como .net, también se admite que se distinguen por el tipo devuelto.
- El reemplazo o sobrescritura se produce cuando se cambia la implementación de un método heredado de la clase padre en la hija.

Las propiedades también pueden presentar polimorfismo. Ejemplos de polimorfismo en propiedades son los tipos genéricos utilizados en Java cuyo tipo se define en tiempo de ejecución, o las variables usadas en PHP o Javascript cuyo tipo depende del contenido de la variable.

Los nombres de las clases son únicos y en teoría no se pueden repetir. Sin embargo, en proyectos complejos en los que trabaja mucha gente, se reutiliza código de otros proyectos o se utilizan librerías de terceros, esto no se puede garantizar. Para evitar conflictos, se definen los espacios de nombres (*namespace*) que agrupan múltiples clases formando una librería y, en principio, son únicos.

La POO resulta muy útil pues permite crear librerías de código y trabajar con ellas de forma independiente. La mayoría de lenguajes que se emplean actualmente en la web lo soportan, por lo que es necesario conocer su uso y práctica.

4.5 Aspectos legales

Los aspectos legales son un elemento ineludible en la realización de proyectos de Internet. Respecto al software objeto de esta memoria, hay que considerar tres aspectos. En primer lugar, la propiedad intelectual de los cursos que utilizan la plataforma, la del *software* de la plataforma y la asociada a las herramientas del estudiante. En segundo lugar, la licencia de uso del programa y los cursos. Y finalmente, el tratamiento de los datos personales de los usuarios de la plataforma en caso de que fuera necesario.

4.5.1 Propiedad intelectual y licencia de uso

En la web del Ministerio de Educación, Cultura y Deporte se define la propiedad intelectual como:

"La propiedad intelectual es el conjunto de derechos que corresponden a los autores y a otros titulares (artistas, productores, organismos de radiodifusión...) respecto de las obras y prestaciones fruto de su creación.

Se considera autor a la persona natural que crea alguna obra literaria, artística o científica. Son objeto de propiedad intelectual todas las creaciones originales literarias, artísticas o científicas expresadas por cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro. La propiedad intelectual de una obra literaria, artística o científica corresponde al autor por el solo hecho de su creación.

La condición de autor tiene un carácter irrenunciable; no puede transmitirse "inter vivos" ni "mortis causa", no se extingue con el transcurso del tiempo así como tampoco entra en el dominio público ni es susceptible de prescripción." [25].⁴³

La legislación española sobre derechos de autor o *copyright* está regulada por la Ley 21/2014, de 4 de noviembre, por la que se modifica el texto refundido de la Ley de Propiedad Intelectual, aprobado por Real Decreto Legislativo 1/1996, de 12 de abril.

Entre otros asuntos, la ley especifica que:

- El autor es quien decide cómo publicar, explotar, reproducir, distribuir o modificar el contenido de su obra.
- El uso de la misma por parte de terceros debe ser autorizado de forma expresa por el autor en los términos que acuerden las partes. Quedan excluidos la copia privada, el uso en la enseñanza y el derecho a la cita.
- La obra queda protegida durante toda la vida del autor y durante los 70 años posteriores a la muerte del mismo en que los derechos pueden transmitirse a sus herederos.

Los derechos del autor se dividen entre morales y patrimoniales.

⁴³ <http://www.mecd.gob.es/cultura-mecd/areas-cultura/propiedadintelectual/la-propiedad-intelectual/definicion.html>

- Los derechos morales son irrenunciables y reconocen al autor de la obra o al ejecutante. Además, reconocen su derecho a exigir respeto a la integridad de la obra y a su no alteración.
- Los derechos patrimoniales están relacionados con el derecho del autor a autorizar o prohibir la explotación de su obra y a cobrar una retribución por ello. Se dividen en:
 - Derechos relacionados con la explotación de la obra.
 - Derechos compensatorios.

Para mejorar la defensa de sus derechos, el autor puede registrar sus obras en el Registro de la Propiedad Intelectual⁴⁴ mientras que las marcas se inscriben en la Oficina Española de Patentes y Marcas⁴⁵.

"[...] La licencia es un documento que expresa la voluntad del autor sobre los límites y alcances del uso que pueden hacer las personas respecto a la copia, reproducción, modificación, traducción y adaptación.

En este sentido se pueden encontrar una variedad de tipos de licencias que pueden ir desde las más restrictivas, en la cual el autor se reserva todos los derechos [...], hasta las licencias más permisivas en las cuales el autor autoriza una amplia posibilidad de usos de la obra a las personas ("algunos derechos reservados"), el extremo de esta posibilidad es el dominio público." [26]

Las críticas a los derechos de autor se centran en que las licencias de uso, impiden el libre acceso, modificación y disfrute de las obras por parte de los usuarios; en especial por el derecho compensatorio por la copia privada, fotocopias de libros, reproducción de música en espacios públicos... Ante estos problemas, han surgido formas alternativas y/o complementarias de aplicar la propiedad intelectual como el *copyleft* o las licencias *Creative Commons*.

"*Copyleft*⁴⁶ es un grupo de licencias cuyo objetivo es garantizar que cada persona que recibe una copia de una obra pueda a su vez usar, modificar y redistribuir el propio trabajo y las versiones derivadas del mismo. Unas veces se permite el uso comercial de dichos trabajos y en otras ocasiones no, dependiendo que derechos quiera ceder el autor." [27]. Ejemplos de *copyleft* son las licencias de software GNU/GPL⁴⁷, *Creative Commons* u otras⁴⁸.

*Creative Commons*⁴⁹ es una organización sin ánimo de lucro que impulsa un sistema flexible de licencias de derechos de autor que permiten a los creadores otorgar a los usuarios permisos para compartir y explotar su obra bajo determinadas condiciones. En ningún caso supone una renuncia al *copyright*, sino se trata de ofrecer una forma simple de crear licencias de uso. Una licencia *Creative Commons* consta del reconocimiento del autor (irrenunciable) y de tres condiciones de explotación opcionales que son:

- Se permite la explotación para usos no comerciales.
- No se permite crear obras derivadas.
- Se autoriza la creación de obras derivadas siempre que estas mantengan la misma licencia que la original.

Es un tema complejo para los no versados, pero que conviene tener en cuenta por las posibles repercusiones legales que pudiera tener.

En este trabajo, los elementos sujetos a derechos de autor son:

⁴⁴ <http://www.mecd.gob.es/cultura-mecd/areas-cultura/propiedadintelectual/registro-de-la-propiedad-intelectual/organizacion-y-direcciones.html>

⁴⁵ <http://www.oepm.es>

⁴⁶ <http://fundacioncopyleft.org/>

⁴⁷ <http://www.gnu.org/home.es.html>

⁴⁸ <http://fundacioncopyleft.org/es/10/licencias>

⁴⁹ <http://es.creativecommons.org/blog/>

- El curso, realizado a partir del manual “Luminotecnia. Iluminación de interiores y exteriores”⁵⁰ y formado por los textos, imágenes y elementos interactivos. El autor es Javier García Fernández y Oriol Boix Aragonès como coautor. Está registrado con el ISBN: 84-600-9647-5.
- Plataforma de *software* para publicar los cursos. Consiste en un *software* cliente-servidor que genera la interface del usuario a partir de documentos HTML e imágenes... la plataforma contiene el curso, y un conjunto de librerías propias y de terceros que le añaden funcionalidad.
- Barra de herramientas del estudiante (*Web Student Tools*). Se trata también de una aplicación cliente-servidor.
- Iconos de la barra de herramientas y diseño de la interfaz gráfica que corresponde a una tercera persona.
- Librerías de código de terceros utilizadas respetando las licencias de uso correspondientes.

4.5.2 Tratamiento de datos personales

En los inicios de Internet, los usuarios se dedicaban prácticamente en exclusiva a consumir información. Añadir contenidos a la web era un privilegio de unos pocos con acceso a un servidor. El resto de personas sólo podían intercambiar contenidos a través del uso del correo electrónico o servicios como *news*⁵¹.

Con la progresiva implantación y popularización de Internet aparecen el comercio electrónico, y nuevas formas y herramientas para gestionar los contenidos. Cada vez hay más información personal en la red que puede llegar a ocasionar perjuicios al usuario, al ser vulnerado su derecho a la intimidad y protección personales (acceso a datos bancarios, datos sobre su identidad, sobre sus gustos o preferencias...).

El usuario, como primer interesado, debe desarrollar un rol activo en la gestión de sus datos, adquiriendo conocimientos y nuevas capacidades que le permitan evitar la invasión de su espacio personal: instalando programas de seguridad como antivirus, anti-spam o cortafuegos, limitando los datos personales que se dan a conocer a terceros, aplicando mecanismos de protección para niños y adolescentes...

El generador de información o servicios también debe velar por la protección de los datos que le suministran los usuarios evitando su robo y uso no autorizado fuera de las condiciones de uso del servicio.

Y por último, las autoridades reguladoras también tienen una responsabilidad hacia los usuarios creando un marco legal adecuado y homogéneo. Así lo recoge la Directiva 95/46/CE del Parlamento Europeo y del Consejo, de 24 de octubre de 1995, relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos.

"(25) Considerando que los principios de la protección tienen su expresión, por una parte, en las distintas obligaciones que incumben a las personas, autoridades públicas, empresas, agencias u otros organismos que efectúen tratamientos -obligaciones relativas, en particular, a la calidad de los datos, la seguridad técnica, la notificación a las autoridades de control y las circunstancias en las que se puede efectuar el tratamiento- y, por otra parte, en los derechos otorgados a las personas cuyos datos sean objeto de tratamiento de ser informadas acerca de dicho tratamiento, de poder acceder a los datos, de poder solicitar su rectificación o incluso de oponerse a su tratamiento en determinadas circunstancias;" [28]

Y su trasposición a la legislación del Estado Español en la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal actualizada el 31/12/2003. En el Título I, Disposiciones Generales, artículos 1 y 2, se establecen el objeto y ámbito de aplicación.

"Artículo 1. Objeto.

⁵⁰ <http://recursos.citcea.upc.edu/llum/llum>

⁵¹ *Network News Transport Protocol* o NTP es un protocolo de Internet creado para la publicación y lectura de noticias. Está descrito en la RFC977 aunque fue sustituida posteriormente por la RFC 3977. [80]

La presente Ley Orgánica tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar.

Artículo 2. Ámbito de aplicación.

1. La presente Ley Orgánica será de aplicación a los datos de carácter personal registrados en soporte físico, que los haga susceptibles de tratamiento, y a toda modalidad de uso posterior de estos datos por los sectores público y privado." [29]

En el artículo 3, Disposiciones Generales, Título I de la citada Ley se definen los conceptos clave que conforman la Ley.

"Artículo 3. Definiciones.

A los efectos de la presente Ley Orgánica se entenderá por:

- a) Datos de carácter personal: cualquier información concerniente a personas físicas identificadas o identificables.
- b) Fichero: todo conjunto organizado de datos de carácter personal, cualquiera que fuere la forma o modalidad de su creación, almacenamiento, organización y acceso.
- c) Tratamiento de datos: operaciones y procedimientos técnicos de carácter automatizado o no, que permitan la recogida, grabación, conservación, elaboración, modificación, bloqueo y cancelación, así como las cesiones de datos que resulten de comunicaciones, consultas, interconexiones y transferencias.
- d) Responsable del fichero o tratamiento: persona física o jurídica, de naturaleza pública o privada, u órgano administrativo, que decida sobre la finalidad, contenido y uso del tratamiento.
- e) Afectado o interesado: persona física titular de los datos que sean objeto del tratamiento a que se refiere el apartado c) del presente artículo.
- f) Procedimiento de disociación: todo tratamiento de datos personales de modo que la información que se obtenga no pueda asociarse a persona identificada o identificable.
- g) Encargado del tratamiento: la persona física o jurídica, autoridad pública, servicio o cualquier otro organismo que, sólo o conjuntamente con otros, trate datos personales por cuenta del responsable del tratamiento.
- h) Consentimiento del interesado: toda manifestación de voluntad, libre, inequívoca, específica e informada, mediante la que el interesado consienta el tratamiento de datos personales que le conciernen.
- i) Cesión o comunicación de datos: toda revelación de datos realizada a una persona distinta del interesado.
- j) Fuentes accesibles al público: aquellos ficheros cuya consulta puede ser realizada, por cualquier persona, no impedida por una norma limitativa o sin más exigencia que, en su caso, el abono de una contraprestación." [29]

En la versión de desarrollo de las herramientas del estudiante sólo se guarda un dato de carácter personal que es el correo electrónico del usuario. Este dato sirve para que el usuario pueda gestionar su perfil y cambiar la contraseña, sus preferencias o darlo de baja. Para proteger los datos de carácter personal se ha implementado un procedimiento de disociación consistente en la encriptación de los mismos, se solicita consentimiento al interesado y se le comunica que sus datos no serán accesibles al público, no serán cedidos y no serán tratados por terceros. Según la LOPD, además, sería necesario crear un fichero y darlo de alta en el registro de la Agencia Española de Protección de Datos o en *l'Autoritat Catalana de Protecció de Dades* (en el caso de ficheros de carácter público).

Dado el carácter experimental del proyecto y el nulo interés lucrativo del mismo, en la versión de producción seguramente se eliminará cualquier dato de carácter personal para no tener que dar de alta el fichero. No obstante, se continuará informando a los usuarios de sus derechos, la política de privacidad y se solicitará su consentimiento para inscribirlo en la web.

5 La tecnología

La intención de este estudio es crear un conjunto de aplicaciones que mejoren la experiencia de aprendizaje de los alumnos. Con esta finalidad, en su diseño, programación y estructura, se emplean tecnologías web, actuales o de aparición inmediata en el mercado. Las herramientas se integran en el conjunto de tecnologías web de segunda generación (Web 2.0) y tercera (Web 3.0)

"El término Web 2.0 comprende aquellos sitios web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la *World Wide Web*. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual, a diferencia de sitios web estáticos donde los usuarios se limitan a la observación pasiva de los contenidos que se han creado para ellos. Ejemplos de la Web 2.0 son las comunidades web, los servicios web, las aplicaciones Web, los servicios de red social, los servicios de alojamiento de vídeos, las wikis, blogs, ..." [30]

"Web 3.0 es una expresión que se utiliza para describir la evolución del uso y la interacción de las personas en internet a través de diferentes formas entre los que se incluyen la transformación de la red en una base de datos, un movimiento social hacia crear contenidos accesibles por múltiples aplicaciones non-browser, el empuje de las tecnologías de inteligencia artificial, la web semántica, la Web Geoespacial o la Web 3D." [31]

A través de este proyecto, los usuarios podrán interactuar con los contenidos y adaptarlos a sus necesidades, ser partícipes de la experiencia formativa y establecer relaciones con otras personas, compartir sus experiencias y beneficiarse de ellas. Objetivos, todos ellos, definidos en los conceptos WEB 2.0 y WEB 3.0.

Se pretende crear las herramientas necesarias para hacerlo posible y que los resultados sean reproducibles y utilizables por cualquier persona con conocimientos mínimos.

5.1 Criterios de selección

En el mercado existen una gran variedad de tecnologías disponibles para crear ricas aplicaciones web interactivas. Para decidir cuáles son las más adecuadas, se van a definir unos criterios razonables, basados en la experiencia.

- a. Que cualquier desarrollador convenientemente preparado pueda adaptar y modificar las soluciones aquí propuestas o crear otras nuevas.
- b. Que utilicen el protocolo HTTP/HTTPS para la comunicación entre el dispositivo del alumno (cliente) y el servidor.
- c. Que funcionen sobre un navegador de internet que respete los estándares del W3C⁵², la ECMA⁵³ u otros organismos similares públicamente reconocidos.

Uno de los mayores problemas para todo desarrollador web son las diferencias entre los navegadores a la hora de implementar los estándares. Esto genera incompatibilidades y alarga el tiempo de desarrollo. Ha habido y siguen existiendo inconvenientes de este tipo, entre versiones de un mismo navegador (Microsoft Internet Explorer o IE) o entre navegadores de diferentes firmas (Netscape 4 contra IE4), (IE contra Mozilla Firefox, Apple Safari, Opera y Google Chrome). Afortunadamente, la tendencia actual de los fabricantes es respetar los

⁵² W3C (*World Wide Web Consortium*). Consorcio internacional encargado de la definición de estándares para la web. <http://www.w3.org/>

⁵³ ECMA (*European Computer Manufacturers Association*). Asociación internacional dedicada a la estandarización de sistemas de información. <http://www.ecma-international.org/>

estándares. Las diferencias se pueden solventar invirtiendo más tiempo de desarrollo, utilizando *frameworks cross-browser*⁵⁴ como jQuery⁵⁵ o restringiendo el desarrollo a algunas versiones y navegadores.

La proliferación actual de soportes (PC, *tablets*, teléfonos inteligentes o *smartphones*, *smartTV*, consolas...) complica más este panorama y hace necesario acotar la compatibilidad. Una correcta evaluación de los usuarios y sus hábitos de navegación puede ayudar a restringir el problema.

De todas maneras, en este proyecto se pretende apostar por los estándares industriales de aceptación mayoritaria y huir de las particularidades de un navegador concreto por buenas que sean sus prestaciones.

- d. Que no se precise de la instalación de complementos en el navegador del usuario (*plug-ins*, *activeX*...). Todas las aplicaciones desarrolladas utilizarán exclusivamente las capacidades nativas del navegador. Se limitan así las necesidades de memoria del dispositivo, de instalar programas suplementarios y se simplifica el desarrollo al reducir el número de tecnologías empleadas. Aun así, las herramientas han de ser lo suficientemente versátiles para poder convivir con páginas web donde sí estén presentes estos contenidos (*flash*, vídeo, sonido, applets de java...). Esto será muy beneficioso en entornos donde es habitual que los dispositivos no admitan la instalación de extensiones o programas como en las aulas informáticas o bibliotecas.
- e. Que sean estándares reconocidos en el mercado.
- f. Que la licencia de uso esté libre de costes que no sean los derivados del aprendizaje de la tecnología utilizada. En la actual situación económica, las instituciones educativas han de reducir costos evitables. Por ello, se defiende el uso de tecnologías de libre uso, que ofrezcan como mínimo el mismo nivel de prestaciones que otras de pago. Es también una forma de garantizar el acceso a la tecnología desarrollada a cualquier institución o individuo independientemente de sus posibilidades económicas.

Quedan excluidos de este principio los costes ocultos, cómo son el disponer de personal especializado (analistas y programadores), y del tiempo y recursos necesarios para la formación de estos.

- g. Que existan herramientas de desarrollo para estas tecnologías, contrastadas, fáciles de utilizar, simples y gratuitas. Si se defiende el uso de tecnologías de libre uso, por su bajo coste, no tiene sentido que sea necesario adquirir un IDE (entorno de desarrollo) para crearlas. Por ello, se emplearán tecnologías para las que exista una gran variedad de programas de desarrollo que se adapten a la forma de trabajar y conocimientos de la organización o los individuos que intervengan.
- h. Que exista una comunidad de desarrollo grande y documentación abundante. De esta manera es posible obtener ayuda para desarrollar el proyecto, documentarse e incluso disponer de módulos de código que ofrezcan funcionalidades útiles en el desarrollo del proyecto.
- i. Que baste con una estructura de *hardware* básica (una máquina de escritorio con una conexión de red) para poner en funcionamiento el proyecto y que se pueda ir ampliando a medida que aumenten las necesidades.

Respecto a la novedad de las tecnologías empleadas, es intención de este estudio aprovechar al máximo las posibilidades que ofrecen los nuevos estándares para web, sin caer en los excesos de tecnologías para las que aún no hay un soporte amplio en el mercado, ni sacrificar prestaciones para así garantizar una mayor compatibilidad con navegadores antiguos.

⁵⁴ *Cross-browser* librería de código que permite al programador desarrollar una aplicación independientemente de las particularidades del navegador sobre el que funciona.

⁵⁵ jQuery. Librería *cross-browser* desarrollada en javascript. <http://jquery.com/>

El carácter de investigación aplicada de este estudio y el hecho de que la instalación de un navegador sea gratuita para los usuarios, justifica en opinión del autor esta apuesta.

5.2 Selección tecnológica

Según las premisas señaladas en el punto anterior se ha elaborado un listado con las tecnologías a emplear en el desarrollo:

- Presentación y formato de los contenidos web
 - HTML
 - CSS
 - *Webfonts*
 - Posicionamiento en buscadores SEO (*Search Engine Optimization*)
- Formatos de imágenes
 - Mapa de bits o *raster*
 - JPEG, JPG
 - GIF
 - PNG
 - Vectoriales. SVG
- Tecnologías de uso transversal en entorno cliente y servidor
 - Codificación de caracteres. Unicode (UTF-8)
 - Expresiones regulares
- Intercambio de información entre el cliente y el servidor
 - JSON
- Almacenamiento de información en el servidor
 - Configuración y diccionarios multilingües para las aplicaciones. XML y JSON
 - Bases de datos relacionales MySQL. Lenguaje SQL
- Modificación dinámica del contenido de las páginas en el entorno cliente. *Client Side*
 - Javascript/ECMA-script
 - DOM y otras APIs HTML
 - AJAX – XMLHttpRequest
- Lenguaje del servidor de aplicaciones web en el entorno servidor. *Server Side*
 - PHP
- Servidor web
 - Apache web server

Como norma general se ha optado por utilizar las recomendaciones y especificaciones oficiales de los lenguajes, descartando las versiones en desarrollo.

En el caso de HTML y CSS, las elevadas prestaciones que ofrecen HTML5 y CSS3, y el hecho de que su implementación en los navegadores esté muy avanzada justifican su uso. El usuario podrá actualizar de manera sencilla su navegador, que es gratuito, para poder disfrutar así de una experiencia mejorada.

Para los programas empleados en el servidor, como PHP o MySQL, se han escogido las versiones estables ya que en los servidores comerciales, las versiones instaladas casi nunca son las últimas disponibles.

5.3 Contenidos de la web

Los contenidos didácticos consisten en un conjunto de archivos almacenados en el servidor como elementos estáticos⁵⁶. Se dividen en:

- Archivos con el texto y explicaciones del curso. Se guardan en formato de texto plano⁵⁷ y están escritos en el lenguaje de etiquetas HTML. Se conectan entre sí mediante enlaces de hipertexto.
- Archivos con el formato (tipografías, colores, estilos de párrafo, etc.). Son archivos de texto plano con instrucciones para aplicar los estilos conocidos como hojas de estilo CSS.
- Archivos multimedia (imágenes, vídeo, audio...).
- Archivos con el código para añadir interacción y efectos especiales escritos en javascript y en formato de texto plano.
- Otros archivos (fuentes de letra...)

Crear contenidos HTML no es un proceso complejo ni requiere de grandes conocimientos. Prácticamente cualquier editor de texto permite guardar archivos en este formato. Eso sí, normalmente es necesario editar el código para optimizarlo, pues los editores suelen añadir una gran cantidad de instrucciones innecesarias.

Analicemos con más detenimiento la producción y el almacenamiento de los contenidos.

Desde el punto de vista de facilitar la implementación del servidor (*server-side*), que no para el creador/ editor de contenidos, lo mejor es utilizar archivos HTML estáticos. No requieren pre-procesamiento de los archivos por parte del servidor ni consumen recursos antes de enviarlos al cliente. El acceso a ellos, es por tanto, más rápido. Para el formador son fáciles de crear, pero con la excepción de los usuarios avanzados el código no será óptimo y tendrá que ser modificado por un técnico si hay que añadir aplicaciones interactivas o mejorar el CSS. Con este sistema, para crear una página ha de intervenir como mínimo una persona que reúna estos dos perfiles: el de redactor y el de técnico.

Para el redactor, es mejor emplear una solución de web 2.0 donde el usuario sólo se preocupa de crear los contenidos de forma transparente a la tecnología empleada. Sin embargo, este tipo de soluciones obliga a crear implementaciones más complejas en el servidor, puesto que hay que programar interfaces de usuario para introducir los textos al estilo de la Wikipedia, WordPress, Twitter, Facebook..., disponer de la infraestructura necesaria (bases de datos u otros sistemas de soporte para almacenar los contenidos) y crear las plantillas para generar las páginas HTML.

Como hemos visto, no existe una solución ideal que satisfaga al formador/usuario. Cada una tiene ventajas e inconvenientes, por lo que es necesario tener muy claro qué es lo que queremos ofrecer y a quien.

En este proyecto se ha optado por una solución mixta que combina archivos HTML estáticos y herramientas 2.0. El formador crea y guarda sus contenidos en formato HTML. Y posteriormente, el servidor se encarga de extraerlos de la página HTML y generar con ellos un nuevo documento con una estructura y estilo predefinidos. Así mismo, se ofrecen una serie de pautas para definir elementos interactivos de una forma simple y accesible para aquellos que no tengan conocimientos del tema.

5.3.1 Lenguaje de etiquetas

HTML es sin lugar a dudas la estrella de Internet y el responsable de su popularización. Desde su creación en 1989 por el CERN⁵⁸ supuso una auténtica revolución respecto a los mecanismos de navegación existentes hasta entonces como

⁵⁶ Un archivo estático es aquel que está almacenado en el disco duro del servidor y que no necesita ser compilado y ejecutado antes de ser enviado al navegador donde se visualizará. Lo contrario es lo que ocurre con las páginas ASP, PHP, JSP...

⁵⁷ “Un archivo de [...] texto plano [...] (en inglés *plain text*) es un archivo informático compuesto únicamente por texto sin formato, sólo caracteres, lo que lo hace también legible por humanos.” [172]

FTP o Gopher⁵⁹. Con HTML aparece la posibilidad de ir directamente de un documento a otro a través de los enlaces de hipertexto.

HTML (*HyperText Markup Language*) es un lenguaje de marcas o etiquetas, no un lenguaje de programación. Sirve para crear documentos de texto con elementos multimedia (gráficos, vídeo y audio) e hiperenlaces a otros documentos. En esencia, es un documento de texto con una serie de etiquetas (*tags*) que sirven para describir la estructura lógica del documento, la distribución y el contenido del mismo. La entidad reguladora es el *World Wide Web Consortium* (W3C). Funciona sobre el protocolo HTTP (*Hypertext Transfer Protocol*)⁶⁰.

La estructura básica de un documento escrito en HTML está formada por una cabecera (*head*) y un cuerpo (*body*).

En la cabecera se coloca la información del documento (especificación de HTML empleada, título...) y los metadatos (autor, fecha de creación, caducidad, tipo de documento...). También se pueden incluir el formato de estilo aplicado al documento (CSS) y *scripts* de código, aunque lo recomendable es ponerlos en archivos separados e incluirlos como recursos externos.

En el cuerpo del documento se encuentra el texto plano convenientemente formateado y maquetado con la ayuda de etiquetas. Estas pueden tener carácter semántico⁶¹, de estructura del contenido⁶², de formato⁶³ o de enlace a recursos externos⁶⁴.

⁵⁸ CERN. Organización Europea para la Investigación Nuclear.

⁵⁹ FTP(*File Transfer Protocol*) es un protocolo que permite acceder y manipular archivos del servidor como si se tratara del explorador de WindowsTM.

Gopher. Servicio que permite el acceso a la información a través de menús organizados en forma de árbol por los que se navegaba para acceder a los documentos.

⁶⁰ "[...] HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC [...] que especifica la versión 1.1. HTTP, define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL). [...]" [103]

⁶¹ Las etiquetas semánticas hacen referencia al contenido de la información por encima de la presentación o la estructura del documento. Las hay puramente semánticas o parcialmente. Por ejemplo <h1> es una etiqueta que introduce el título de la página aunque tiene asignada el estilo negrita y una fuente muy grande; el resto de etiquetas <hX> sirve para definir el título de las diferentes secciones de la página. indica que el contenido es importante y que tiene que ir en negrita. En HTML5 hay nuevas etiquetas que permiten definir mejor las secciones de un documento como <article> o <section>.

⁶² Etiquetas como <div> o se utilizan básicamente para maquetar contenidos. No tienen valor semántico ni aportan formato.

⁶³ En este grupo se incluirían etiquetas como <p> para indicar párrafos, texto en negrita, y para listas,...

⁶⁴ Etiquetas como , <object>, <a>,... permiten añadir recursos multimedia que residen en archivos externos o enlazar unas páginas con otras.

```

<!DOCTYPE html>                                <!-- versión de HTML -->
<html>                                           <!-- Indica el inicio del documento -->
<head>                                           <!-- Inicio de la cabecera -->
  <title>Mi primera página</title>              <!-- título del documento -->
  <link rel="stylesheet" type="text/css" href="estilo.css" /> <!-- recurso externo CSS -->
  <script type="text/javascript" src="codigo.js"></script> <!-- recurso externos script -->
</head>                                         <!-- Final de la cabecera del documento -->
<body>                                          <!-- Inicio del cuerpo del documento -->
  <h1>Mi primera página</h1>                   <!-- título tamaño 1 -->
  <p>Esto es un párrafo.</p>                   <!-- título tamaño 1 -->
</body>                                        <!-- Final del cuerpo del documento -->
</html>                                        <!-- Final del documento -->

```

Código 6 Estructura básica un documento HTML

El lenguaje HTML se deriva de SGML⁶⁵. Actualmente existen dos líneas de desarrollo:

- HTML
- XHTML

En el primer caso, la última versión oficial es la 4.01 aunque su sucesor HTML 5 está en una fase muy avanzada de desarrollo e implementación y en buena parte ya está soportado por los navegadores actuales. Su uso está cada vez más extendido ya que ofrece unas prestaciones muy superiores a HTML4. En el segundo caso, XHTML está en la versión 1.1. Sintácticamente son muy similares, aunque XHTML es más riguroso con la sintaxis de las etiquetas. Consigue así garantizar su compatibilidad con XML (por ejemplo, es obligatorio que todas las etiquetas tengan apertura y cierre) y que el documento esté bien formado. [32]

HTML5 tiene una naturaleza dual, pues soporta tanto notación tipo HTML como XML. De esta manera se garantiza su compatibilidad con las APIs que trabajan con archivos XML.

En el contenido del curso utilizado en este proyecto se ha optado finalmente por HTML5 aunque inicialmente se pensó en XHTML 1.1. Además, se han empleado algunas de las APIs⁶⁶ de HTML5 para añadir funcionalidad al material educativo.

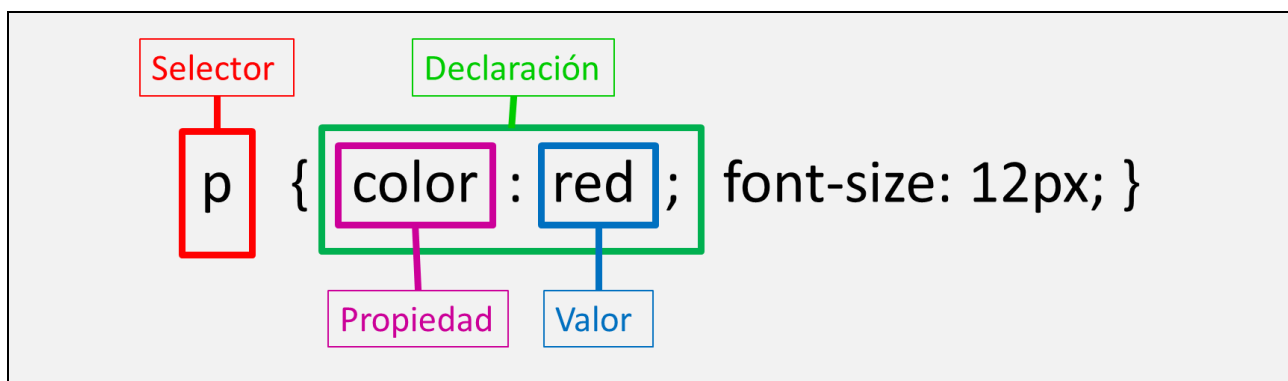
5.3.2 Estilo de los documentos

Una de las grandes aportaciones que acompañaron, en su momento, a la aparición de HTML4 respecto a versiones anteriores fue la separación del contenido de la presentación, siendo esta completamente personalizable gracias a las hojas de estilo en cascada o CSS (*Cascading Style Sheets*). Esta novedad facilitó el mantenimiento de las aplicaciones web y permite trabajar independientemente a los creadores de contenidos y los diseñadores.

CSS se basa en el uso de reglas sintácticas que permiten personalizar el aspecto visual de las etiquetas HTML.

⁶⁵ “The Standard Generalized Markup Language (ISO 8879:1986 SGML) is an ISO-standard technology for defining generalized markup languages for documents [...] Generalized markup is based on two [...] postulates: [...] Markup should be declarative: it should describe a document's structure and other attributes, rather than specify the processing to be performed on it. [...] Markup should be rigorous so that the techniques available for processing rigorously-defined objects like programs and databases can be used for processing documents as well.” [66]

⁶⁶ API (*Application Programming Interface*) Interfaz de programación de aplicaciones. Es un conjunto de funciones que ofrece una librería de software para ser utilizado desde el código de otro programa.



Código 7 Definición del estilo en una etiqueta p

El ejemplo anterior ilustra cómo se define un estilo. Primero se especifica el elemento HTML afectado con un selector. Luego se escribe la lista de reglas de estilo o declaraciones aplicadas (formada por parejas propiedad-valor). En este caso, se aplicará a los párrafos una fuente de tamaño de 12 píxeles y color rojo.

Existen muchos tipos de propiedades que permiten controlar el aspecto visual de la página. Se pueden clasificar en:

- Características tipográficas
- Párrafos
- Fondos
- Visualización de tablas y listas
- Maquetación

La recomendación de CSS se estructura en niveles, donde cada nuevo nivel engloba las propiedades del anterior para guardar la compatibilidad. Actualmente está en vigor CSS2 aunque ya está en fase de desarrollo avanzado CSS3 y algunos aspectos de CSS4. El grado de soporte de CSS3 por parte de los navegadores es desigual, aunque muchas propiedades ya se pueden utilizar.

En este proyecto se utiliza básicamente CSS2 aunque se incluye CSS3 en algunas aplicaciones para mejorar la presentación (como en la definición del estilo de las tablas).

Otra de las ventajas de CSS es que permite utilizar tipos de letras que residan en el servidor o *webfonts*. De esta manera, todos los usuarios ven las páginas igual independientemente de las tipografías que tengan instaladas en su ordenador. Así, cuando un usuario solicita una página HTML al servidor, este le envía el contenido junto al resto de archivos entre los que se encuentran las fuentes. Por problemas de compatibilidad entre los navegadores existen diferentes formatos de archivo para las tipografías. [33]

5.3.3 Accesibilidad

La accesibilidad consiste en un conjunto de técnicas que mejoran el acceso de los usuarios a las páginas web tanto si padecen una discapacidad, como por cualquier otra causa ya sea el acceso con una tecnología anticuada, un dispositivo móvil, una conexión lenta, etc.

De entre las guías desarrolladas con esta finalidad por la *Web Accessibility Initiative* o WAI, las más importantes para el desarrollador web son la WCAG⁶⁷ y la ARIA⁶⁸.

⁶⁷ <http://www.w3.org/TR/WCAG20/> [53]

⁶⁸ <http://www.w3.org/TR/wai-aria/> [174]

La WCAG explica cómo desarrollar contenidos web accesibles. En su versión dos de consta de doce guías formadas por reglas que se organizan en cuatro partes o principios:

- Que la información sea **perceptible**. Es decir, que no haya información invisible a todos los sentidos del usuario.
- Que la interfaz sea **operable** independientemente del dispositivo de entrada que utilice el usuario.
- Que la información y el uso de la interfaz sean **comprensibles** (claras y precisas).
- Que el contenido sea accesible de forma independiente a la tecnología empleada por el usuario ahora o en el futuro; que sea **robusto**.

Son reglas que obedecen al sentido común. A modo de ejemplo vamos a mencionar las siguientes:

- Utilizar tipografías claras y fáciles de leer
- Procurar que el índice de contraste de color CCR entre el fondo y la letra sea superior a 4,5
- Estructurar el documento con la ayuda de etiquetas semánticas y de cabecera hX
- Marcar el idioma de la página
- Utilizar un lenguaje breve, conciso, claro y fácil de leer
- Poner un atributo *alt* en todas las imágenes que no sean decorativas
- Etcétera

ARIA ayuda a las tecnologías de soporte y a los navegadores informando de cómo se estructura la página web y qué papel juegan los elementos interactivos en ella. Realiza tres funciones:

- Define cómo es la estructura semántica de la página (formularios, contenido principal, anuncios, barras de navegación...). Esto se consigue añadiendo un atributo llamado *role* a las etiquetas HTML.
- Especifica qué función realiza cada uno de los elementos interactivos de una aplicación web que utilice un lenguaje de *script*. Por ejemplo un menú desplegable, una galería de imágenes... Para ello se combina el atributo *role* con otros atributos llamados propiedades (*aria-propiedad*).
- Se garantiza que se puede visualizar y acceder a todas las aplicaciones de una página web y a todas sus funciones independientemente del dispositivo de entrada. Esto se consigue con una combinación de HTML, CSS y javascript.

```
<ul role="menubar">
  <!-- Rule 2A: "File" label via aria-labelledby -->
  <li role="menuitem" aria-haspopup="true" aria-labelledby="fileLabel">
    <span id="fileLabel">File</span>
    <ul role="menu">
      <!-- Rule 2C: "New" label via Namefrom:contents -->
      <li role="menuitem">New</li>
      <li role="menuitem">Open...</li>
      ...
    </ul>
  </li>
  ...
</ul>
```

Código 8 Ejemplo de uso de atributos ARIA en un menú construido con una lista no ordenada⁶⁹

5.3.4 Posicionamiento en buscadores

El posicionamiento en buscadores, más conocido por sus siglas en inglés SEO (*Search Engine Optimization*) es un conjunto de técnicas y recomendaciones que permiten mejorar la posición de un sitio web. De esta manera se facilita el acceso a la información relevante del sitio web y es más fácil captar nuevos usuarios.

⁶⁹ Font: http://www.w3.org/TR/wai-aria/roles#tac_example1_header

En su mayoría, son técnicas que apelan al sentido común: las hay que se refieren al contenido, la navegación, la estructura de la web, facilitar el trabajo del robot de búsqueda y mejorar la reputación de la publicación.

Entre las técnicas referidas al contenido se destacan:

- Cada página ha de tener un título único que sea descriptivo y breve
- Ha de tener una etiqueta `<meta name="description">` única que resuma el contenido de forma concisa pero breve
- El contenido ha de seguir una estructura jerárquica esquemática donde cada sección vaya acompañada de su correspondiente etiqueta `<hX>`
- Utilizar etiquetas con valor semántico
- Ofrecer contenidos originales, exclusivos y de calidad
- Utilizar un lenguaje sencillo
- Acompañar las imágenes de textos descriptivos
- Emplear palabras clave
- Etcétera

El título y la descripción son importantes, pues son los textos que aparecen en el listado de resultados del buscador.

De las técnicas para facilitar la navegación y la estructura mencionaremos:

- Utilizar una jerarquía de árbol en que la raíz sea la página de inicio
- Utilizar un número limitado de niveles de navegación
- Añadir a cada página la ruta completa (técnica de las migas de pan o *breadcrumb*)
- Uso de URL semánticas o amigables⁷⁰ que incorporen las palabras clave
- Incorporar un mapa de la web
- Escribir un texto descriptivo en los enlaces
- Destacar los enlaces
- Personalizar la página de error 404 (página no encontrada)
- Utilizar una estructura de carpetas simple

Para facilitar el trabajo de los robots tenemos:

- Añadir el archivo robots.txt que da instrucciones al robot sobre que partes del sitio web puede indexar. [34]
- Añadir un archivo sitemap.xml con la lista de URL del sitio web que puede explorar el robot. Se puede ofrecer información extra como la fecha de última actualización, frecuencia de los cambios o la importancia de la página en relación a las demás. [35] [36]
- Indicar con el atributo `rel="nofollow"` aquellos enlaces que no debe seguir

La reputación de la publicación se mide por el número y prestigio de las páginas que enlazan con nuestro sitio web, la presencia de enlaces provenientes de redes sociales, la originalidad del contenido, la relación con webs de temática similar, etc.

Finalmente, hay que recordar que conviene evitar todas aquellas prácticas que penalizarían la posición del sitio web entre los resultados. [20]

⁷⁰ Son aquellas direcciones web que resultan legibles para los seres humanos. Contienen palabras mayoritariamente en lugar de números y parejas clave-valor. Por ejemplo, una web como la siguiente <http://www.ejemplo.com/index.php?seccion=sociedad&articulo=000123> se sustituiría por algo como: <http://www.ejemplo.com/sociedad/los-famosos-no-estan-de-moda>

5.3.5 Alternativas

Una alternativa posible al uso de HTML en la distribución de documentación electrónica es el uso de PDF [37]. Aunque inicialmente era un formato propietario, ahora es un estándar abierto.

En PDF, es fácil generar documentos, y actualmente los editores más populares permiten guardar archivos en este formato. Entre sus ventajas está que en un único archivo se puede incluir todo lo necesario para su visualización como textos, imágenes, vídeo, estilos, tipos de letra, scripts, etc. y que la impresión será tal y como se ve en la pantalla.

Aunque su uso está muy extendido, básicamente se utiliza para crear documentos simples o formularios. El resto de posibilidades que ofrece el formato como la interactividad, está actualmente infrautilizado, o es desconocido y/o ignorado por los usuarios.

Una desventaja es que los navegadores no disponen de soporte nativo, hay que abrirlo con un programa externo o utilizar un complemento.

Por todo esto, se ha optado por no utilizarlo como soporte para los contenidos de los cursos.

5.4 Formatos de imágenes

Los formatos de imágenes pueden dividirse en dos grandes familias:

- Mapa de bits o *raster*
- Vectoriales

La diferencia primordial radica en la manera en que se guarda la información en el archivo. En las primeras, la imagen se reduce a una matriz de puntos o píxeles cada uno de los cuales lleva asociado un color.

En las segundas, se guardan formas geométricas junto a una serie de características que las definen como son el color, la posición, la forma... Otra diferencia radica en que las imágenes vectoriales permiten cambiar el zoom o deformar la imagen sin que se produzcan pérdidas de calidad en el visionado.

5.4.1 Mapa de bits o *raster*

Los mapas de bits son imágenes formadas por píxeles individuales dispuestos en una rejilla rectangular de x por y píxeles con una profundidad de z colores; por ejemplo, para los GIF z puede tener un valor de 256, 16 o 2 colores y para JPEG z es de 16,7 millones de colores. Para reducir el tamaño de los archivos, la información se almacena utilizando algoritmos de compresión. Los hay de dos tipos: los que provocan pérdidas de calidad y los que no.

En los formatos sin pérdidas, la paleta de colores de la imagen se guarda completa. Mientras que en los que tienen pérdida se asimilarán varios colores muy similares entre sí a uno solo. A menor calidad más tonalidades de un color se asimilarán al mismo tono, consiguiendo una paleta final más reducida. Si la calidad es alta, se emplean muchos más colores y llega un punto en que el ojo humano no nota la diferencia debido a sus limitaciones fisiológicas.

El principal defecto de estos formatos es el pixelado. Cada imagen tiene un tamaño y una definición. Si, por ejemplo, intentamos ampliar una imagen y pasar de unas dimensiones de 100x100 (10000) píxeles a 200x200 (40000) píxeles hay que rellenar 30000 píxeles más. La pregunta, ahora, es como hacerlo si sólo tengo información para una cuarta parte. Se pueden hacer dos cosas. Una es aumentar el tamaño de los puntos (pasan a ser puntos gordos) y otra conservar el tamaño de los puntos e interpolar la información que falte. En cualquiera de los dos casos llega un momento que esto no es posible sin que se aprecie la transformación (imagen degradada).

Si, por el contrario, se reduce la imagen, aparece un problema similar aunque con los algoritmos adecuados es más fácil de disimular.



Ilustración 14 Imagen original.



Ilustración 15 Imagen ampliada donde se aprecia la pérdida de calidad causada por el pixelado.

En el ámbito de las aplicaciones web los formatos para imágenes más extendidos y por tanto los que utilizaremos son:

- GIF
- PNG
- JPEG

GIF (*Graphics Interchange Format*) es uno de los primeros formatos empleados en las páginas web. Es adecuado para almacenar dibujos y animaciones. Sus características principales son:

- Algoritmo de compresión sin pérdida
- Profundidad de color máxima de 256 u 8 bits
- Soporta paletas entre 2 y 256 colores

- Tamaño de archivo muy reducido
- Canal *alpha*, se puede definir un color como transparente
- Soporte de animación por *frames*

PNG (*Portable Network Graphics*) nació como alternativa no sujeta a patentes de GIF (aunque actualmente éste ya no tiene la patente en vigor). Es adecuado para dibujos estáticos, imágenes con pocos colores o sustituir a los GIF no animados. Características:

- Algoritmo de compresión sin pérdida
- Profundidad de color máxima, color verdadero
- Soporta paletas de colores reducidas
- Canal *alpha*, soporta transparencia
- Tamaño de archivo inferior a GIF

JPEG, JPG (*Joint Photographic Experts Group*) es un formato adecuado para imágenes con gran cantidad de colores como las fotos. Características:

- Algoritmo de compresión con pérdida
- Se puede escoger el nivel de calidad
- El tamaño del archivo varía según el nivel de calidad
- Profundidad de color, color verdadero

5.4.2 Vectorial

La principal ventaja de los formatos vectoriales es que si se deforma la imagen o se cambia de tamaño esta mantiene su aspecto y no sufre el efecto, no deseado, del pixelado.

En Internet cada vez está más difundido el uso de SVG (*Scalable Vector Graphics*), por ser un estándar abierto. SVG deriva del lenguaje XML y consiste en un archivo de texto plano escrito en un lenguaje de marcas similar a HTML. Posteriormente, el navegador convierte el código en una imagen de mapa de bits que es lo que se visualiza en pantalla. Es adecuado sobre todo para sustituir a imágenes con poca profundidad de color como los GIF y PNG.

El hecho de que la imagen se reduzca a un conjunto de etiquetas en lugar de a un archivo binario abre la puerta a interesantes aplicaciones como es el caso de los gráficos interactivos. La creación y manipulación de imágenes SVG de forma dinámica es mucho más sencilla que la de los formatos *raster*, tanto desde el lado del servidor como desde el cliente.

Características:

- Soporte nativo en los navegadores actuales
- Escalable
- El contenido puede ser indexado por un buscador
- Permite incluir formas geométricas vectoriales, imágenes de mapas de bits y texto
- Admite animaciones simples con SMIL⁷¹ o lenguajes de *script*
- Permite interacción utilizando lenguajes de *script*, DOM y eventos
- Se puede comprimir con *gzip*

⁷¹ SIML. “[...] *Synchronized Multimedia Integration Language* (lenguaje de integración multimedia sincronizada) [...] estándar del [...] W3C para presentaciones multimedia. [...] permite integrar audio, vídeo, imágenes, texto o cualquier otro contenido multimedia.” [68]

Otro formato vectorial que se utiliza actualmente es *Flash* de Adobe (SWF). El formato SWF comenzó siendo propietario aunque en la actualidad es un formato abierto. Algunas características son:

- Necesita un complemento para su ejecución aunque su uso y soporte está muy extendido
- Interactivo
- Soporta el lenguaje de *script* ActionScript
- Escalable
- Puede incluir objetos vectoriales, imágenes *raster*, audio y vídeo

En los últimos años ha caído en desuso en favor de otros formatos como SVG, HTML5 que incluye soporte nativo para audio y vídeo o CSS3 que permite crear animaciones con elementos de la página. Flash aún se utiliza mucho para la elaboración de vídeos animados o *banners* publicitarios. No obstante se ha descartado por quedar fuera de los criterios de selección expuestos (no está soportado de forma nativa por el navegador).

5.5 Tecnologías transversales

Se han agrupado en este apartado tecnologías que por su naturaleza se aplican tanto en la capa cliente como en la del servidor de la aplicación. No son exclusivas de un ámbito y su uso es generalista. Son:

- Codificación de caracteres, Unicode y UTF8.
- Expresiones regulares.

5.5.1 Codificación de caracteres

La codificación de caracteres es una tecnología que permite representar mediante glifos⁷² los caracteres de los sistemas de escritura u otros símbolos en la pantalla o en cualquier otro medio electrónico. A cada glifo se le asigna un código que lo identifica. El problema es que mientras la relación entre glifo y código es unívoca (un glifo – un código), al revés no se cumple (1 código – 1 o más glifos).

Glifo	Código carácter (Hexadecimal)	Página de código
a	61	ISO 8859-1 / Latin-1
Á	C1	ISO 8859-1 / Latin-1
Б	C1	Windows-1251 ⁷³
ع	C1	ISO/IEC 8859-6 ⁷⁴

Tabla 1 Ejemplos de codificación de caracteres

En los primeros ordenadores, debido a las limitaciones físicas de los sistemas, la memoria disponible para almacenar los códigos era muy limitada. Se definieron 128 símbolos (siete bits y se añadió un bit de paridad para verificar errores) correspondientes a los caracteres usados en el idioma inglés, los números y otros símbolos especiales como signos de puntuación, retorno de carro, no imprimibles... Se les conoce como juego de caracteres ASCII [38] y forma parte del núcleo de todos los demás juegos. Con el tiempo se amplió el número de códigos disponibles, pero seguía siendo insuficiente para representar todos los glifos de todas las lenguas existentes y además, no era práctico.

⁷² “En tipografía, un glifo es una representación gráfica de un carácter, de varios caracteres o de parte de un carácter y es el equivalente actual del tipo de imprenta (la pieza que tenía grabada la letra). Un carácter es una unidad textual mientras que un glifo es una unidad gráfica.” [107]

⁷³ Windows-1251. Página de códigos empleada para almacenar caracteres cirílicos. [162]

⁷⁴ ISO/IEC 8859-6. Página de códigos empleada para caracteres arábigos. [154]

Dec ⁷⁵	Hex ⁷⁶	Carácter	Dec	Hex	Carácter	Dec	Hex	Carácter	Dec	Hex	Carácter
0	0	NUL ⁷⁷	32	20	SP	64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS ⁷⁸	40	28	(72	48	H	104	68	h
9	9	HT ⁷⁹	41	29)	73	49	I	105	69	i
10	00A	LF ⁸⁰	42	02A	*	74	04A	J	106	06A	j
11	00B	VT ⁸¹	43	02B	+	75	04B	K	107	06B	k
12	00C	FF	44	02C	,	76	04C	L	108	06C	l
13	00D	CR ⁸²	45	02D	-	77	04D	M	109	06D	m
14	00E	SO	46	02E	.	78	04E	N	110	06E	n
15	00F	SI	47	02F	/	79	04F	O	111	06F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	01A	SUB	58	03A	:	90	05A	Z	122	07A	z
27	01B	ESC	59	03B	;	91	05B	[123	07B	{
28	01C	FS	60	03C	<	92	05C	\	124	07C	
29	01D	GS	61	03D	=	93	05D]	125	07D	}
30	01E	RS	62	03E	>	94	05E	^	126	07E	~
31	01F	US	63	03F	?	95	05F	_	127	07F	DEL

Tabla 2 Caracteres ASCII (0-127)

La solución fue adaptar la página de códigos a cada lengua, país, cultura, etc. A los caracteres ASCII se añadían los particulares de cada una de estas. Así se formaban nuevas páginas como la ISO 8859-1 para Europa occidental o Shift JIS para el japonés. Esto dificultaba extraordinariamente la internacionalización de los programas. Era habitual que un texto o programa que utilizara caracteres que no pertenecieran al inglés, no se visualizara bien en otros sistemas diferentes a aquel para el que había sido creado.

⁷⁵ Código decimal del carácter.

⁷⁶ Código hexadecimal del carácter.

⁷⁷ Valor nulo.

⁷⁸ Espacio en blanco.

⁷⁹ Tabulador horizontal.

⁸⁰ Nueva línea.

⁸¹ Tabulador vertical.

⁸² Retorno de carro

Texto	Codificación
El Barça és un gran club.	ISO 8859-1
El Bar❖a ❖s un gran club.	UTF8
El Barηa ιs un gran club.	ISO 8859-7 (griego)
El Barⴰ ⵓs un gran club.	ISO 8859-6(árabe)
El Bar 軋 駿 un gran club.	Shift JIS (japonés)

Tabla 3 Ejemplo de caracteres ilegibles por problemas de codificación

Con internet el problema se hizo más evidente pues la audiencia pasó de ser local a global. Para lograr una visualización correcta de la web es imprescindible especificar la página de código con que fue creada y que el ordenador del usuario la reconozca. En los años 90 del siglo XX aparece Unicode⁸³ como una iniciativa para resolver este problema.

"Unicode es un juego (*set*) de caracteres universal, es decir, un estándar en el que se definen todos los caracteres necesarios para la escritura de la mayoría de los idiomas hablados en la actualidad que se usan en la computadora. Su objetivo es ser [...] un superconjunto de todos los sets de caracteres que se hayan codificado."
[39]

Los caracteres Unicode se pueden codificar con 8 bits (UTF8), 16 bits (UTF16) y 32 bits (UTF32). Esto supone que se pueden representar hasta 2^{32} caracteres; o lo que es lo mismo casi 4295 millones de glifos.

En UTF8, los glifos tienen longitud variable y se utilizan de uno a cuatro bytes⁸⁴ por carácter. Es compatible con US-ASCII de siete bits y su uso está muy extendido en internet (páginas y correo electrónico).

"La distribución de caracteres es la siguiente:

- Caracteres codificados con un byte: Los incluidos en US-ASCII, un total de 128 caracteres.
- Caracteres codificados con dos bytes: Un total de 1920 caracteres. Este grupo incluye los caracteres romances más signos diacríticos, y los alfabetos griego, cirílico, copto, armenio, hebreo, árabe, siríaco y Thaana entre otros.
- Caracteres codificados con tres bytes: Caracteres del plano básico multilingüe de Unicode, que unido al grupo anterior, incluye la práctica totalidad de caracteres de uso común, entre ellos los caracteres del grupo CJK: chino, japonés y coreano.
- Caracteres codificados con cuatro bytes: Caracteres del plano suplementario multilingüe. Símbolos matemáticos y alfabetos clásicos para uso principalmente académico: Lineal B silábico e ideográfico, alfabeto persa, fenicio... Y el plano suplementario ideográfico: caracteres Han de uso poco común.

" [40].

Haciendo trabajar a todas las capas de la aplicación web con la misma página de códigos, facilitamos que se entiendan entre ellas y que la visualización sea correcta independientemente de lengua empleada por el servidor o el cliente. Para ello son necesarios los siguientes pasos:

- Capa cliente
 - Los archivos, a nivel de bytes, han de estar codificados en UTF8.
 - Cada archivo HTML ha de incluir una cabecera *meta* que indique que utiliza UTF8.
- Capa servidor

⁸³ The Unicode Consortium. <http://www.unicode.org/>

⁸⁴ 1 byte = 8 bits

- La BD ha de estar codificada en UTF8. MySQL permite especificar esto a nivel de gestor de BD, base de datos, tabla ...
- El servidor web ha de configurarse para enviar archivos en UTF8.
- El servidor de aplicaciones también ha de configurarse para generar archivos en UTF8.
- Hay que tener especial cuidado en la conexión del servidor de aplicaciones con la BD. Si el gestor de BD no trabaja en UTF8, aunque la BD sí, pueden haber problemas. De igual manera, si se emplea un conector que no sea nativo como ODBC, JDBC, .net... hay que asegurarse que trabaje en UTF8.

En su defecto habrá que hacer las conversiones necesarias de páginas códigos para que todo funcione bien.

5.5.2 Expresiones regulares

Una expresión regular es un patrón de caracteres que se utiliza para realizar búsquedas o hacer comparaciones dentro de una cadena de caracteres o *string*. Uno de sus usos habituales es validar el formato de los datos introducidos por el usuario. La sintaxis puede variar según el motor de comparación utilizado aunque básicamente existen dos:

- POSIX o `ereg`. Es la más antigua de las dos.
- PCRE o `preg`. Expresiones regulares compatibles con PERL que se ha convertido en el estándar de facto por su mayor versatilidad y potencia.

En este proyecto se utilizan las segundas porque son soportadas por todas las aplicaciones (Apache, MySQL) y lenguajes de programación (Javascript, PHP, SQL) empleados en el proyecto.

<p>Expresión:</p> <p style="text-align: center;"><code>"^[\\dXY]\\d{7}[A-Z]\$"</code></p> <p>Significado:</p> <p>El primer carácter puede ser un número (0-9) <code>"\\d"</code> o una de las letras "X o Y" mayúsculas</p> <p>Los siguientes siete caracteres son números <code>"\\d{7}"</code></p> <p>El último carácter es una letra del conjunto "A-Z"</p> <p>Nota:</p> <p>La expresión regular no comprueba la validez de la letra final</p>

Código 9 Ejemplo de PCRE para validar un DNI/NIE

5.6 Intercambio de información entre el cliente y el servidor

Dado que en el proyecto se apuesta por implementar un cliente pesado⁸⁵ basado en AJAX y la arquitectura MVC⁸⁶ en la que el cliente y el servidor intercambiarán principalmente datos y no páginas HTML completas, se hace necesario definir cómo será este intercambio. Entre las opciones consideradas están:

- JSON
- XML
- formatos propietarios definidos por el usuario
- PHP serializado

⁸⁵ Un cliente pesado solicita al servidor los datos para procesarlos en la aplicación cliente que genera la presentación. Por el contrario, un cliente ligero realiza unas pocas operaciones y es el servidor el que realiza casi toda la carga de cálculo y genera la presentación.

⁸⁶ MVC. Modelo Vista Controlador.

5.6.1 JSON

JSON⁸⁷ es un formato ligero para intercambio de datos entre aplicaciones independiente de la plataforma. Aunque está basado en javascript/ ECMA-262⁸⁸, actualmente tiene su propia especificación ECMA-404⁸⁹ y MIME, RFC4627⁹⁰.

Entre sus principales características están:

- Simplicidad
- Es legible para los seres humanos
- El código es fácil de generar e interpretar para una máquina
- Es independiente del lenguaje de programación de las aplicaciones
- Es texto plano

Su uso está muy extendido actualmente y hay implementaciones de librerías JSON en la gran mayoría de lenguajes de programación (PHP, javascript, C#, Ruby, Java, VB...)

Las estructuras de datos en JSON se basan en la representación de:

- Objetos, estructuras... como colecciones de parejas clave-valor
- Los vectores o arreglos (*arrays*) como listas ordenadas de valores

La sintaxis es muy simple:

- Los valores pueden ser números, *string* Unicode entrecomillados (""), true, false, null, un objeto o un *array*
- Los objetos son listas entre {} de la forma nombre: valor separados por ','
- Los vectores son listas entre [] de valores separados por ','
- Objetos y vectores pueden anidarse

```
<?php
class miClase{
    public $a;
    public $b;
    public $c;

    public function suma() { return $this->a + $this->b + $this->c;}
}
$var = new miClase();
$var->a = 1;
$var->b = 3;
$var->c = 4;
$var2= array(
    $var,
    "Mi casa",
    -36.35,
    true
);
echo "<pre>".json_encode($var2)."</pre>";
/*
resultado de convertir a JSON el objeto $var2
[{"a":1,"b":3,"c":4},"Mi casa",-36.35,true]
*/
?>
```

Código 10 Ejemplo de código que trabaja con JSON

⁸⁷ <http://www.json.org/>

⁸⁸ ECMA-262. Documento donde se describe la sintaxis de javascript [43].

⁸⁹ ECMA-404. Documento donde se describe la sintaxis de JSON [67]

⁹⁰ RFC4627. [61; 153]

La comodidad y la facilidad para implementar JSON como formato de intercambio de datos lo hacen idóneo para su uso en este proyecto.

5.6.2 XML

XML⁹¹ es un lenguaje desarrollado a partir del SGML por el W3C para almacenar datos y facilitar su intercambio entre aplicaciones. Los datos se guardan en archivos de texto plano, según estructuras definidas por el usuario. Las reglas sintácticas para describir la estructura son estrictas con los errores. Se dice que un documento está bien formado si no tiene errores y que es válido si su estructura sigue las reglas definidas en un documento DTD o XSD.

Los datos van entre etiquetas similares a las de HTML con la forma `<nombre_etiqueta></nombre_etiqueta>` y se pueden añadir atributos para almacenar información adicional. El nombre y el significado de las etiquetas lo especifica el programador libremente.

```
<?xml version="1.0" encoding="UTF-8"?>
<dictionary language="es">
  <error>
    <dbDelete>ERROR. No se ha podido completar la operación de borrado de la BD.</dbDelete>
    <dbSave>ERROR. No se ha podido guardar la nota en la BD.</dbSave>
    <defaultCategory>Important</defaultCategory>
    <aboveUnderline>ERROR. No se permite superponer subrallados.</aboveUnderline>
  </error>
  <messages>
    <options>Opciones</options>
    <save>Guardar</save>
    <minimize>Minimizar</minimize>
    <maximize>Maximizar</maximize>
    <close>Borrar</close>
  </messages>
</dictionary>
```

Código 11 Ejemplo de archivo XML

Las características que lo hacen especialmente interesante para su aplicación son:

- Es extensible. Una vez definido se pueden añadir nuevas etiquetas
- Es independiente de la plataforma
- Soporta internacionalización y plurilingüismo
- Es compatible con Unicode
- Es libre y no hay que pagar licencias por utilizarlo
- Es un estándar
- Es modular
- Es fácil de leer y manipular con cualquier editor de texto

Otra gran característica de XML es que forma parte de una familia de tecnologías diseñadas para realizar tareas habituales con los datos.

- XML Schema/XSD. Lenguaje basado en XML que sirve para definir la estructura del documento XML; qué elementos, atributos y tipos de datos pueden contener, y como están organizados
- DTD. Document Type Definition. Se utiliza también para definir la estructura. Sustituido por XSD
- XSL/XSLT. Lenguaje de hojas de estilo que permite definir transformaciones para convertir datos de archivos XML en otro tipo de documentos como, por ejemplo, XHTML
- XPath y Xpointer son lenguajes para localizar y acceder a partes y datos de un documento XML
- Xlink es un lenguaje de direccionamiento que permite crear enlaces bidireccionales entre documentos

⁹¹ [52], [51], [106]

- XQuery y XQL son lenguajes para acceder a los datos mediante consultas
- XML DOM. API para acceder y manipular XML desde un lenguaje de programación
- XHTML. Adaptación de HTML para que pueda ser procesado como XML
- SOAP (Simple Object Access Protocol). Protocolo basado en XML para intercambiar información a través de HTTP
- WSDL (*Web Services Description Language*). Lenguaje para describir servicios web
- RDF (*Resource Description Framework*). Lenguaje para describir recursos en la web
- SVG. Formato de imágenes vectoriales basado en XML
- XUA. Agente de usuario de XML. Estandarización de navegadores para que soporten las especificaciones de XML.
- XML-Signature Xades
- XML-Encryption
- Etcétera

XML se utiliza, entre otros usos, como formato de almacenamiento de información alternativo a las bases de datos. Es muy práctico si los datos no necesitan modificaciones frecuentes. Los archivos XML resultan cómodos y bastante seguros de manipular con un editor, sobre todo para aquellos usuarios finales no familiarizados con los lenguajes de programación. Además, tanto PHP como javascript ofrecen soporte nativo para la lectura y escritura de los archivos XML. Para crear y editar XML se pueden emplear editores especializados o un simple editor de texto. Su principal inconveniente radica en la complejidad para acceder a los datos y manipular la información cuando se programa el código fuente. Alternativas como JSON o SQL son mucho más prácticas y sencillas de usar.

Aunque XML se puede utilizar para transmitir información entre el cliente y el servidor se ha descartado en favor de JSON, mucho más sencillo de manipular y tratar.

5.6.3 Alternativas

Una alternativa es definir un formato propietario creado expresamente para el proyecto. Tiene la ventaja de que se ajustaría perfectamente al propósito del programa, pero se ha descartado porque no cumple el criterio de ser un estándar industrial.

Otra sería utilizar PHP serializado. Este formato permite convertir un objeto de una clase cualquiera en un *string* con formato donde se almacena el nombre, el tipo de dato y el valor de las propiedades del objeto. Una de sus principales ventajas es que si disponemos de la definición de la clase, se recupera el objeto directamente sin necesidad de crear una función que introduzca los datos JSON o XML en él. Cada vez es más popular y ya existen implementaciones en otros lenguajes como Java, que permiten recuperar datos enviados en este formato.

```
class miClase{
    public $a;
    public $b;
    public $c;

    public function suma() { return $this->a + $this->b + $this->c;}
}

$var = new miClase();
$var->a = 1;
$var->b = 3;
$var->c = 4;
echo "<pre>".serialize($var)."</pre>";

/*
resultado de serializar el objeto $var
O:7:"miClase":3:{s:1:"a";i:1;s:1:"b";i:3;s:1:"c";i:4;}
*/
```

```
Explicación:
Objeto de la clase miClase (7 letras) con 3 propiedades
Propiedad 'a' (string 1 letra) → valor 1 (entero)
Propiedad 'b' (string 1 letra) → valor 3 (entero)
Propiedad 'c' (string 1 letra) → valor 4 (entero)
*/
```

Código 12 Ejemplo de PHP serializado

Que sea dependiente del lenguaje del servidor de aplicaciones (PHP), que no sea un estándar y que el lenguaje del cliente, javascript, no tenga una librería para trabajar con él han contribuido a que no considere acertado su uso en este proyecto.

5.7 Persistencia de la información

Antes de profundizar en este apartado, hay que distinguir en cómo se organiza la información en el proyecto. Existen cuatro capas de gestión.

- Capa de los contenidos educativos
- Capa de la interfaz gráfica GUI
- Capa de gestión de usuarios
- Capa de contenidos adaptados por el usuario

Tal como se ha expuesto anteriormente, los contenidos educativos serán tratados como contenidos estáticos y guardados en archivos HTML e imágenes que procesará una herramienta para generar el aspecto definitivo. Así mismo, se desarrollarán aplicaciones para incorporar interactividad y se utilizará la librería Mathjax⁹² para mostrar fórmulas científicas.

La capa de la interfaz gráfica incluye los archivos de configuración de las aplicaciones, los textos que aparecen en estas, los mensajes de error y otros textos de soporte. Como la finalidad es que la aplicación soporte multilenguaje, y para facilitar el mantenimiento, se ha optado por tratarlos como contenidos estáticos JSON.

La capa de gestión de usuarios sólo se implementa a modo de demostración, pues lo ideal sería que la aplicación estuviera integrada en un campus virtual. Es un contenido dinámico y los datos se almacenarán en una base de datos relacional.

La capa de contenidos adaptados del usuario es el núcleo del proyecto y se considera también un contenido dinámico por su alta volatilidad. Aunque es posible almacenar información en el cliente, esta función se reserva para el servidor por los motivos que se expondrán a continuación.

Otros datos, como los archivos de configuración de la aplicación se implementarán en PHP por motivos de seguridad.

5.7.1 Almacenamiento en el cliente

El almacenamiento de datos en el navegador del cliente es un asunto controvertido porque afecta a la privacidad de las personas. De por sí, la posibilidad de poder identificar a los usuarios es buena porque permite la existencia de

⁹² <https://www.mathjax.org/>. Librería para mostrar fórmulas en todos los navegadores. Soporta fórmulas escritas en notación TeX, AsciiMath y MathML.

sesiones⁹³ con las que podemos, por ejemplo, comprar o leer el correo, personalizar las aplicaciones según sus gustos y necesidades, y realizar un seguimiento de las visitas para extraer conclusiones que mejoren los servicios prestados. Pero el mal uso y los abusos que han realizado algunos proveedores de servicios con sus prácticas han acabado generando rechazo hacia esta tecnología.

Hasta la aparición de HTML5, el único sistema de almacenamiento de datos disponible eran las *cookies*. Una *cookie* es una porción de información almacenada como texto plano que se almacena en el disco duro del usuario. Los datos que se guardan son: el nombre de esta, el valor con la información, los datos del servidor que la creó, la fecha de caducidad y otras características de seguridad.

```
Name    OAX
Value    JM/PFlHLpRkABC+O
Host     .miweb.cat
Path     /
Expires  Fri, 11 Mar 2016 18:08:59 GMT
Secure   No
HttpOnly No
```

Código 13 Ejemplo de contenido de una *cookie*

El uso de *cookies* presenta los siguientes inconvenientes:

- El tamaño y la cantidad de *cookies* que puede crear un servidor en un navegador está limitada por las especificaciones (RFC6265 [41]).
- Las *cookies* se envían al servidor como texto plano en la cabecera de la [petición HTTP](#), y esto supone un potencial problema de seguridad.
- No constituyen un sistema fiable de almacenamiento. El usuario puede rechazar su uso o borrarlas en cualquier momento.

Por estos motivos, su papel se reserva únicamente a guardar los datos de configuración de la aplicación como el idioma preferido o para la identificación del usuario mientras dure la sesión de trabajo.

HTML5 dispone de nuevas API para almacenar información:

- *Storage*
- *IndexedDB*

Storage es un mecanismo de almacenamiento interno del navegador que sólo es accesible desde javascript. Dispone de dos interfaces; una para datos permanentes de la aplicación, *localStorage*, y otra para datos temporales, *sessionStorage*, que sólo se mantienen mientras esté abierta la sesión con el servidor. La información almacenada sólo es accesible por la aplicación que la creó.

La segunda API es *IndexedDB*. No es una base de datos (BD) relacional propiamente dicha, sino más bien una colección o almacén de objetos (registros) que funcionan como tablas. En cada tabla se define una clave primaria para identificar los objetos y adicionalmente otros índices para facilitar las búsquedas. Además dispone de un conjunto de funciones para realizar las búsquedas, manipular los datos y crear la estructura de la BD.

El uso de *cookies*, *Storage* o *IndexedDB* conlleva los siguientes inconvenientes:

- El usuario es el responsable último de conservar la información y puede decidir borrarla en cualquier momento
- Los datos residen en el dispositivo del usuario

⁹³ Debido a que el protocolo HTTP carece de estado, la única manera que el servidor reconozca al cliente mientras visita las páginas del sitio es que el cliente se identifique cada vez que abre una página. Una de las maneras de hacerlo es mediante *cookies*.

- Se renuncia a la portabilidad de los datos entre diferentes dispositivos que ofrecen las aplicaciones en la nube
- Existe la posibilidad de que el navegador del usuario no soporte las nuevas API de HTML5 con lo cual no funcionaría la aplicación

Estos problemas inherentes a los tres métodos dificultan el control sobre el progreso de los alumnos, y por lo tanto, su uso se reservará para tareas de soporte de la aplicación.

5.7.2 Almacenamiento en el servidor

El almacenamiento de datos en el servidor parece ser la solución más conveniente. Algunas de sus ventajas son:

- Permite el control y seguimiento del progreso de los alumnos por parte del docente; algo fundamental en la formación en línea
- Accesibilidad a la aplicación 24 horas al día
- Portabilidad de las aplicaciones entre diferentes dispositivos
- Adaptabilidad de la aplicación a las características del dispositivo del usuario. La UI contempla la diversidad de dispositivos desde los que se puede acceder a los contenidos (PC, *tablets*, *Smartphone*, TV...) y es escalable en función de las posibilidades que ofrezca el dispositivo

Como mayor inconveniente, hay que señalar que es necesario dotar al servidor de una mayor infraestructura. Hay que montar sistemas de copias de seguridad (*backup*), implementar la seguridad (antivirus, cortafuegos...), conexiones seguras SSL/TLS⁹⁴, etc... Para hacerlo posible son necesarios conocimientos técnicos avanzados, pero no es un obstáculo insalvable si se contrata o se dispone de un servidor externo que ofrezca todos estos servicios.

Para guardar los datos de la capa de contenidos adaptados, la mejor solución es emplear una base de datos relacional.

"Una base de datos BD (en inglés, *database* DB) es un conjunto de datos o información pertenecientes a un mismo contexto y almacenados sistemáticamente para su uso posterior" [42]. Las bases de datos ofrecen una manera eficiente y práctica de almacenar y manipular grandes cantidades de información.

Existen diferentes tipos de bases de datos en función de cómo se almacenan la información.

- Relacionales. Son las más utilizadas y las que emplearemos en este proyecto
- Espaciales. Adecuadas para almacenar información geográfica (GIS)
- XML
- Orientadas a objetos
- Almacenes de datos (*datawarehouse*)
- Etcétera

Para trabajar con las BD se utilizan programas llamados Sistemas Gestores de Bases de Datos o SGBD que se encargan de:

- Gestión de almacenamiento. Cómo se almacena la información a nivel físico
- Gestión de consultas. Cómo se interpretan y ejecutan las consultas para introducir, modificar, borrar y consultar datos
- Motor de reglas. Cómo se interpretan las consultas a nivel físico
- Gestión de la seguridad. Acceso, usuarios y permisos

Ejemplos de SGBD son MySQL, PostgreSQL, Oracle DB, MS SQL Server, MS Access, etc.

⁹⁴ "Secure Sockets Layer -SSL [...]- y su sucesor Transport Layer Security -TLS [...]- son protocolos criptográficos que proporcionan comunicaciones seguras por una red [...]." [58]

Conceptualmente trabajan a dos niveles:

- A nivel físico, se encargan de gestionar la información contenida en los archivos del disco duro. Optimizan el almacenamiento para minimizar las necesidades de espacio y maximizar la velocidad de acceso.
- A nivel lógico, se encargan de traducir las peticiones de los usuarios y el sistema en procesos que se puedan ejecutar a nivel físico. Las instrucciones se escriben en un lenguaje independiente de la manera en cómo se almacena la información. El usuario solo debe describir qué información solicita, dónde hay que buscarla y qué condiciones ha de cumplir.

En una base de datos relacional la información se agrupa en entidades o tablas que representan objetos del mundo real. Una entidad está formada por registros o filas que son los objetos individuales que queremos guardar. Por cada registro guardamos una serie de datos característicos de interés para la aplicación que se conocen como atributos o columnas. Para poder distinguir las filas unívocamente, se define la clave primaria formada por uno o varios atributos, o alternativamente una clave artificial definida por el diseñador. Las entidades se relacionan entre ellas definiendo relaciones de parejas entre la clave primaria de una tabla y una columna llamada clave foránea en la otra.

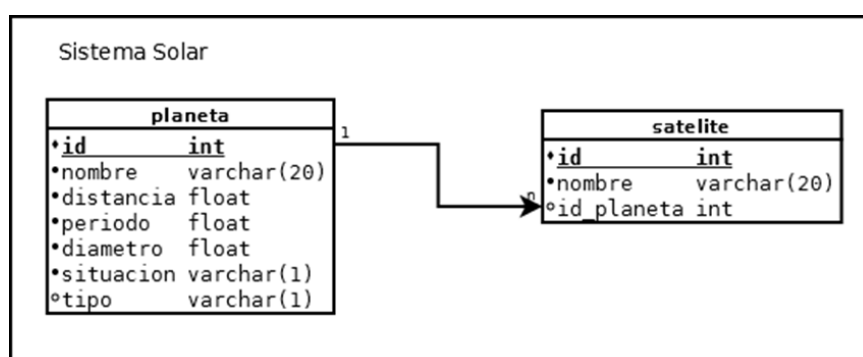


Ilustración 16 Ejemplo de BD relacional

El lenguaje que se emplea para tratar la información se llama SQL (*Structured Query Language*). Es un lenguaje declarativo y de programación que permite manipular y obtener información de una base de datos y actualizarla. El usuario especifica la información que quiere obtener y dónde conseguirla sin preocuparse de cómo se accede a ella físicamente. Aunque el SQL es un estándar ISO⁹⁵, muchos SGBD soportan el SQL con extensiones propias distintas del lenguaje estándar.

SQL se divide en varios lenguajes:

- DDL (*Data Description Language* o Lenguaje de Descripción de Datos) que sirve para crear y manipular las bases de datos, tablas, vistas, etc
- DML (*Data Manipulation Language* o Lenguaje de Manipulación de Datos) que sirve para consultar y modificar los datos
- Lenguaje de programación de *scripts*
- DCL (*Data Control Language* o lenguaje de control de datos) que sirve para gestionar usuarios y permisos
- TCL (*Transaction Control Language* o lenguaje de control de transacciones) que sirve para gestionar transacciones

En este proyecto se utilizará MySQL con licencia GPL como SGBD por su eficiencia, por la experiencia personal de muchos años trabajando con este producto y porque durante mucho tiempo ha estado ligado a PHP; otro de los pilares del proyecto. De forma alternativa se podía haber empleado PostgreSQL o cualquier otro SGBD de *software* libre disponible en el mercado, pero la experiencia con él ha resultado un factor decisivo para su elección.

⁹⁵ ISO/IEC 9075, SQL:2898 / 2011

5.8 Modificación dinámica del contenido de las páginas en el cliente.

Buena parte de las utilidades para que los alumnos puedan adaptar los contenidos del curso a sus necesidades se basa en crear aplicaciones que funcionen total o parcialmente en el lado del cliente (*client-side*). En la comunicación del cliente con el servidor (*server-side*) se utilizará una arquitectura AJAX que actuará en segundo plano y se encargará de gestionar el tratamiento, envío y recepción de los datos. Para programar el código se emplea un lenguaje de *script* conocido como javascript.

Javascript es un lenguaje interpretado que se compila y ejecuta en tiempo real sobre un navegador u otro entorno que disponga del intérprete. La implementación del intérprete es la principal fuente de incompatibilidades entre navegadores ya que sus prestaciones han ido evolucionando con el tiempo y las versiones del lenguaje.

Es un lenguaje funcional orientado a objetos (POO) que no utiliza el mecanismo de herencia clásico padre-hijo como Java, C++, C# o PHP. En su lugar, se añaden los métodos y propiedades al prototipo⁹⁶ de los objetos que se transfiere de unos a otros siguiendo una cadena⁹⁷.

Javascript es el nombre con el que actualmente se conoce a una familia de tecnologías para crear aplicaciones web en el lado del cliente. Estas son:

- Núcleo (*core*). Especificaciones del lenguaje ECMA-script. ECMA-262 [43]. Incluye
 - Sintaxis
 - Objetos *Object*, *Array*, *Boolean*, *Number*, *String*, *Date*, *Math*, *RegExp*, *Error*, *Function*, *Global* y *JSON*
- DOM (*Document Object Model*). API para manipular HTML [44]
- XMLHttpRequest⁹⁸
- XML API⁹⁹
- HTML5 APIs (*Storage*, *Canvas*, *Drag and Drop*, *SVG*, *Stream*, *TextTrack*, *IndexedDB*, *File*, *History*...)
- Etcétera

DOM, *Document Object Model*, es una API regulada por el W3C que permite manipular un documento HTML o XML. Se pueden introducir dinámicamente cambios en los estilos del documento (CSS) o las etiquetas (HTML). Por ejemplo, se pueden mover elementos de la página, modificar contenidos, añadir interactividad, etcétera.

AJAX, *Asynchronous JavaScript And XML*, es un conjunto de tecnologías que permiten actualizar contenidos web de forma asíncrona¹⁰⁰ sin necesidad de volver a cargar la página. La transferencia de datos entre cliente y servidor se realiza en segundo plano y de forma transparente para el usuario. Las tecnologías que intervienen son:

- HTML/ XHTML y CSS para generar las interfaces de usuario GUI
- Javascript para gestionar la comunicación cliente-servidor y controlar al resto de elementos
- DOM para manipular los contenidos
- XMLHttpRequest para realizar la comunicación asíncrona cliente-servidor

⁹⁶ Prototipo. Propiedad de un objeto que contiene la definición del objeto con sus propiedades y métodos.

⁹⁷ Cadena de prototipos. Mecanismo de herencia por el que cuando se invoca una propiedad o método, si este no existe, se busca en el objeto anterior hasta llegar al objeto raíz *Object*. Esto ahorra código puesto que se pueden incorporar nuevas propiedades y métodos a un ancestro y que todos los demás los "hereden".

⁹⁸ <http://www.w3.org/TR/XMLHttpRequest/>

⁹⁹ http://www.w3schools.com/dom/dom_intro.asp

¹⁰⁰ "Comunicación asíncrona. Es la conexión que se establece entre el cliente y el servidor que permite la transferencia de datos no sincrónica, o sea el cliente puede realizar varias peticiones al servidor sin necesidad de esperar por la respuesta de la primera.

A diferencia, las conexiones sincrónicas solo pueden realizar una petición al servidor y hasta que esta no sea respondida no se puede realizar la siguiente." [131]

- XML, JSON, HTML preformateado, texto plano... para transferir los datos
- Servidor aplicaciones para gestionar la petición y generar la respuesta

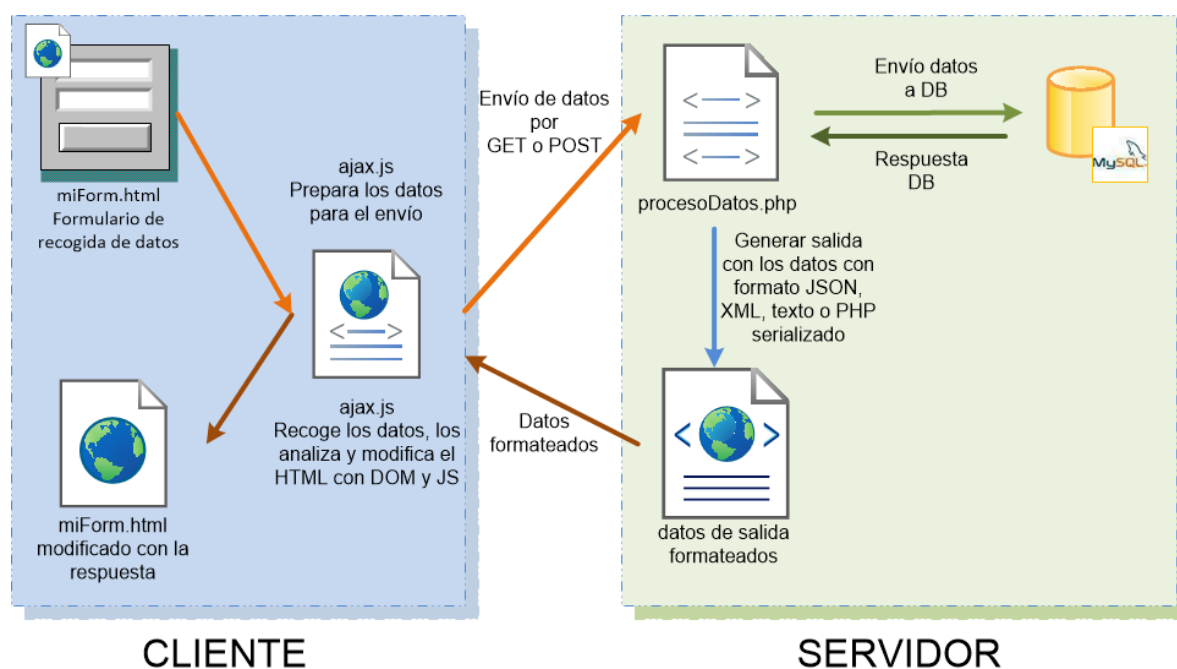


Ilustración 17 Comunicación cliente-servidor con AJAX

Javascript puede utilizarse de forma nativa, cosa que suele presentar problemas de compatibilidad *cross-browser* entre navegadores, o emplear alguno de los *frameworks* o librerías de código como jQuery que permiten simplificar la producción de aplicaciones.

Después de analizar las prestaciones de jQuery se ha decidido descartar su uso por los siguientes motivos:

- Las funcionalidades ofrecidas (funciones gráficas, animaciones, etc.) no resultan de utilidad al proyecto
- Las capacidades *cross-browser* se garantizan solo para las últimas versiones de los navegadores¹⁰¹
- Su uso supone un incremento en el consumo de recursos de la máquina, como la memoria o el tiempo de ejecución que no son compensadas por las ventajas que aporta

Se ha optado por programar código nativo de javascript porque permitirá optimizar los recursos, sobre todo en dispositivos con capacidades limitadas. La intención es que las aplicaciones sean compatibles con el mayor número posible de navegadores, dispositivos y sistemas.

5.9 Server Side y servidor de aplicaciones

Un servidor de aplicaciones es un entorno integrado para desarrollar y ejecutar el modelo de negocio de una organización mediante aplicaciones cliente-servidor. Aunque este concepto es muy amplio, en nuestro caso se va a limitar a aplicaciones que utilicen el protocolo HTTP.

La idea es la siguiente, el servidor web recibe una petición de un cliente que no puede atender porque no existe el recurso estático¹⁰² solicitado; por ejemplo la ficha bibliográfica de un libro. Para generar la respuesta, necesita ejecutar un programa que consulte la base de datos. Entonces, delega el trabajo en el servidor de aplicaciones que se encarga de

¹⁰¹ Tal como se menciona en la página *Browser Support* [127].

¹⁰² En principio, los servidores web solo pueden leer archivos residentes en el disco duro y enviarlos al cliente.

aplicar la lógica de negocio, obtiene los datos y genera la salida en formato HTML. Por motivos de eficiencia, la salida no se guarda en disco, sino que se mantiene en memoria y se envía directamente al servidor web para que cree la respuesta HTTP que envía al cliente. Por supuesto que la salida admite cualquier otro formato o MIME además del HTML como imagen, audio, vídeo, documentos...

En los primeros servidores web, los contenidos dinámicos eran generados por programas ejecutables escritos en C++, Java, Perl o cualquier otro lenguaje; eran los conocidos como CGI¹⁰³. El servidor web pasaba al programa los datos recibidos del cliente y este los procesaba y generaba la respuesta. Esto presentaba algunos inconvenientes:

- Eran potencialmente inseguros. El programa necesita permisos de ejecución y/o escritura y lectura en la carpeta desde donde se ejecuta. Existe el riesgo de que por una incorrecta configuración de los permisos, un usuario malintencionado pudiera subir un archivo dañino y tomar el control del servidor.
- Cualquier modificación del código fuente implica volver a compilar el programa y subirlo al servidor.
- La sintaxis de los lenguajes de programación hace muy complicado y tedioso implementar la salida.
- Cada vez que se ejecuta el programa, implica que el SO¹⁰⁴ ha de arrancar la aplicación, reservar la memoria necesaria, ejecutarlo, etc. Esto incrementa el consumo de recursos en el servidor y empeora el tiempo de respuesta.

Con el tiempo, esta tecnología ha ido evolucionando para superar sus limitaciones. Ejemplos de esto son Java EE¹⁰⁵, C# o Visual Basic .net. Por ejemplo, con el uso de JEE, el programa o *servlet* se carga solo una vez en memoria cuando se inicia el servidor y cada vez que se conecta un cliente se ejecuta un nuevo hilo o *thread*¹⁰⁶. Para facilitar la generación de la salida y la creación de interfaces de usuario se dispone de tecnologías adicionales como JSP, JSF, notación EL...

La otra posibilidad es utilizar lenguajes interpretados para generar las páginas. En este caso, el programa se encuentra en un archivo fuente donde se combina código HTML e instrucciones en el lenguaje de programación empleado. Cada vez que se necesita ejecutar el programa, el intérprete lee el archivo fuente, lo analiza sintácticamente, lo compila y lo ejecuta. El proceso es algo más lento que en el caso anterior, pero su mantenimiento es mucho más sencillo pues basta con modificar el archivo fuente. El intérprete se puede instalar como una librería (preferido por su mayor seguridad) o como un CGI.

A nivel de prestaciones no hay gran diferencia pues ambos ofrecen más o menos las mismas funcionalidades. El uso de un sistema u otro dependerá del software disponible en el proveedor de servicios y por lo tanto, constituye un tema ajeno al objeto de la tesis.

¹⁰³ CGI. *Common Gateway Interface* o Interfaz de entrada común. "Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME" [99].

¹⁰⁴ Sistema Operativo

¹⁰⁵ JEE. *Java Enterprise Edition* [95]. Es una plataforma de programación para desarrollar aplicaciones distribuidas multicapa, basada en componentes, que se ejecutan sobre un servidor de aplicaciones. Incluye entre otras:

- API: JDBC, RMI, JavaMail, JMS, JNDI, Servicios Web, XML, etc.
- Especificaciones para crear componentes:
 - *Servlets*, componentes web escritos en Java.
 - JSP, páginas web dinámicas que combinan HTML y código al estilo de PHP o ASP.
 - EJB, componentes para utilizar en la arquitectura de n-capas.
- Especificaciones para crear módulos distribuibles (war, jar, ear).

¹⁰⁶ Los hilos o *threads* son entornos de ejecución independientes que comparten los recursos de un proceso (memoria, acceso a archivos, red). Todo proceso tiene al menos un hilo.

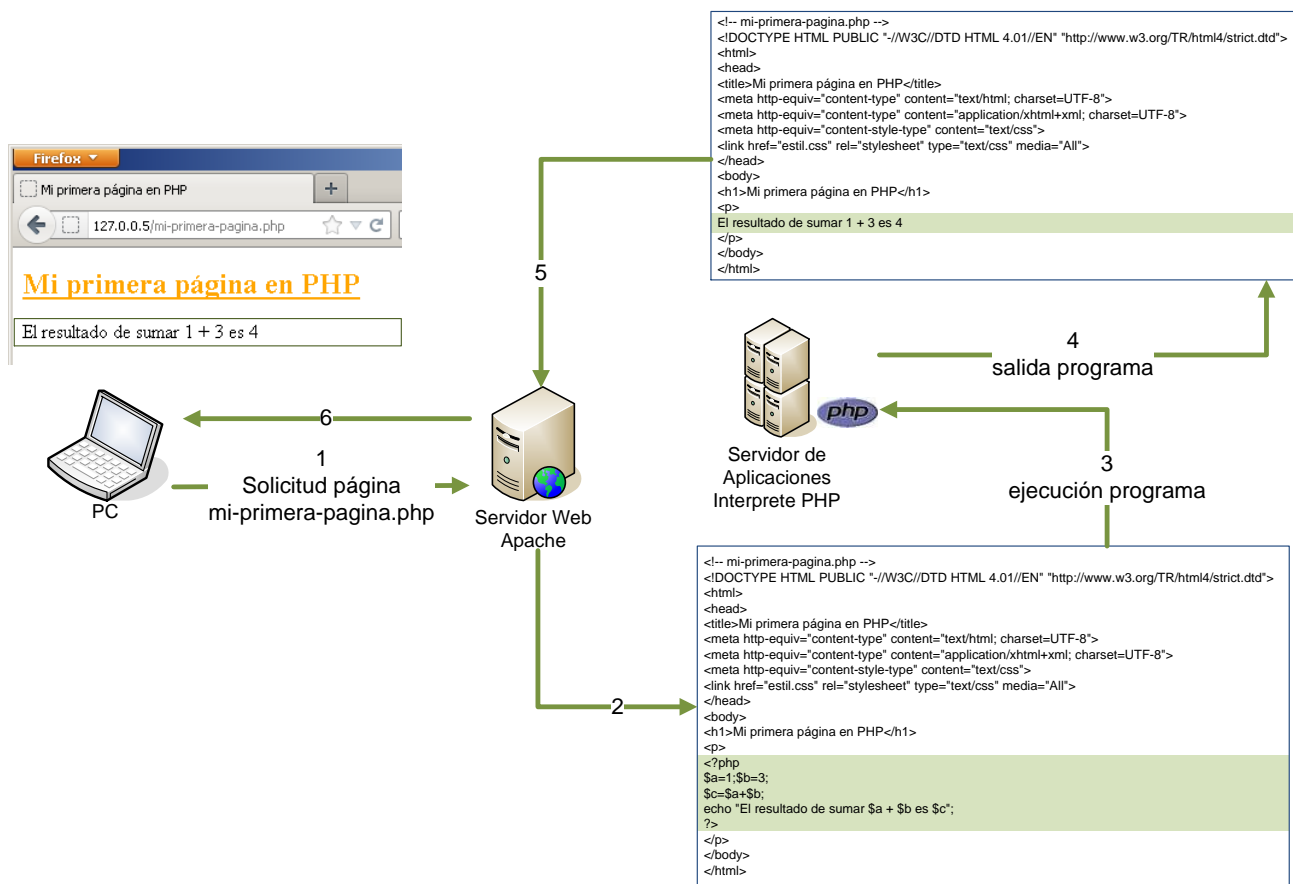


Ilustración 18 Funcionamiento del servidor de aplicaciones para PHP

En el mercado existen muchas tecnologías disponibles para desarrollar aplicaciones web en el servidor. Un primer grupo lo forman los lenguajes compilados como C++, C#, VB, Java... Son lenguajes estructurados, de sintaxis rígida, muy potentes pero con una curva de aprendizaje elevada. No son adecuados para un programador neófito que no tenga claros conceptos avanzados como la programación orientada a objetos (POO u OOP).

Por otro lado tenemos el grupo de los lenguajes interpretados como PHP, ASP .net, Java JSP, Java JSF, Ruby on Rails, Perl... Estos lenguajes resultan más sencillos de aprender y utilizar. Su sintaxis es más sencilla y está muy enfocada a su uso en proyectos web. Para expresar al máximo su potencial hay que aplicar paradigmas de programación avanzada como POO. Para el presente proyecto se ha optado por PHP por ser una tecnología muy popular, potente y flexible en la que el autor acumula muchos años de experiencia como programador y formador.

PHP es un lenguaje nacido para desarrollar aplicaciones web que, con el tiempo, ha ido evolucionando para terminar por convertirse en un lenguaje de uso general que también permite crear aplicaciones de línea de comandos o escritorio. Es muy versátil y potente. Como virtudes podemos destacar:

- Permite mezclar HTML y código PHP
- Está disponible para los principales SO y servidores web del mercado
- Tiene licencia de *software* libre
- Soporta programación orientada a procedimientos y a objetos. Se observa una evolución con los años del primer tipo al segundo que se ha acentuado a partir de la versión 5
- Es modular y dispone de numerosas librerías y funcionalidades adicionales para cubrir las necesidades más comunes en aplicaciones web. Acceso a múltiples BD, XML, servicios web, archivos...

- Sintaxis relajada. El tipo de las variables cambia según las necesidades del programa¹⁰⁷, puede generarse código en tiempo de ejecución, los objetos son extensibles¹⁰⁸...
- Soporta diferentes arquitecturas web como MVC
- Curva de aprendizaje rápida, especialmente para desarrolladores que vienen de lenguajes con sintaxis basada en C
- Dispone de una amplia comunidad de usuarios y un excelente soporte técnico
- Existen numerosas extensiones y *frameworks* para desarrollar aplicaciones disponibles en la web
- Su uso está muy extendido en internet

Y como debilidades:

- Sin una arquitectura y metodología de programación adecuada, el código producido puede resultar caótico y difícil de mantener
- Su sintaxis relajada puede resultar desesperante para programadores que vengan de lenguajes fuertemente estructurados como Java, C++ o .net

Después de valorar sus virtudes y debilidades se ha considerado PHP es la mejor opción para desarrollar las aplicaciones dinámicas en el servidor.

5.10 Servidor web

Es un tipo de *software* de servidor que utiliza el protocolo HTTP para realizar la comunicación con el cliente. Atiende las peticiones y envía como respuesta la información requerida. En su papel de mero intermediario, se limita a transferir los archivos requeridos sin realizar ninguna interpretación. A menudo se instala conjuntamente con un servidor de aplicaciones para dotarlo de la capacidad de generar contenidos dinámicos.

El mercado actual de servidores está formado por un reducido grupo de actores [45]:

- Apache HTTP server
- Microsoft Information Server
- Nginx
- Google web server
- Otros

Apache es el servidor más popular de la lista y llegó a tener una cuota de mercado del 70% aunque esta ha disminuido en la actualidad. Es asimismo, la base de otros proyectos como *Google web server*, *IBM Websphere*, *Apache Tomcat*... Es un *software* libre, desarrollado por una comunidad de usuarios gestionados por la fundación Apache. Es multiplataforma, modular, flexible y ofrece un gran rendimiento para la gestión de contenidos dinámicos. Como punto negativo, decir que es más lento que otros y que la configuración se realiza modificando un archivo de texto.

Microsoft Information Server es un servidor web exclusivo para sistemas operativos Windows con soporte para .net. Soporta múltiples protocolos (al contrario que Apache), dispone de una interfaz amigable y las versiones básicas soportan un número limitados de usuarios concurrentes. No se considera conveniente su uso en este proyecto ya que no es un servidor multiplataforma.

¹⁰⁷ Los tipos se infieren y van cambiando con el tiempo. Esto es una gran ventaja por la flexibilidad que aporta, pero a su vez es un problema porque si no se tiene cuidado puede provocar errores en tiempo de ejecución.

¹⁰⁸ Los objetos se pueden modificar una vez creados añadiendo propiedades o métodos.

Nginx es un servidor multiprotocolo, multiplataforma, de código abierto y muy eficaz para gestionar sitios web con tráfico muy elevado. Actualmente está en alza, pero su uso se ha descartado ya que no es fácil encontrar servidores de alojamiento que lo tengan instalado, como sí ocurre en el caso de Apache.

La elección del servidor web, por lo tanto, ha sido sencilla. Apache es un servidor fiable, muy utilizado en la industria del *hosting* y con el que el autor posee una gran experiencia de uso y configuración.

Es muy importante mejorar el rendimiento de la aplicación, disminuyendo los tiempos de carga. Para ello, es imprescindible una correcta configuración del servidor, que ayudará también, positivamente, en su posicionamiento en los buscadores (SEO). El archivo `.htaccess` es un archivo de configuración local de Apache HTTP Server que se aplica a la carpeta donde se aloja y sus descendientes. Por motivos de seguridad, las directivas de configuración que se pueden aplicar están limitadas. Como ejemplo, se mencionan las siguientes:

- Se pueden definir páginas para personalizar errores
- Redirección de documentos
- Fijar la caducidad de los elementos para la caché de los navegadores
- Compresión de archivos para su envío al cliente
- Reescritura de URL. Entre otros usos permite convertir URL en versiones amigables; por ejemplo convertir *querystrings* de la forma `www.miweb.com/aplicación/index.html?a=casas&b=pera` en otras del estilo `www.miweb.com/aplicación/casas/pera`

6 Curso de pruebas

Para probar el curso se ha utilizado un curso en línea existente sobre iluminación que desarrollé para mi proyecto final de carrera en 1999. El curso se llama *Luminotecnia. Iluminación de interiores y exteriores* y está registrado con el ISBN: 84-600-9647-5. Originalmente se encontraba en la dirección <http://edison.upc.edu/curs/llum>, pero actualmente está en <http://recursos.citcea.upc.edu/llum>. El curso ha sido un gran éxito y, con el tiempo, se ha convertido en una referencia sobre el tema en lengua castellana¹⁰⁹.

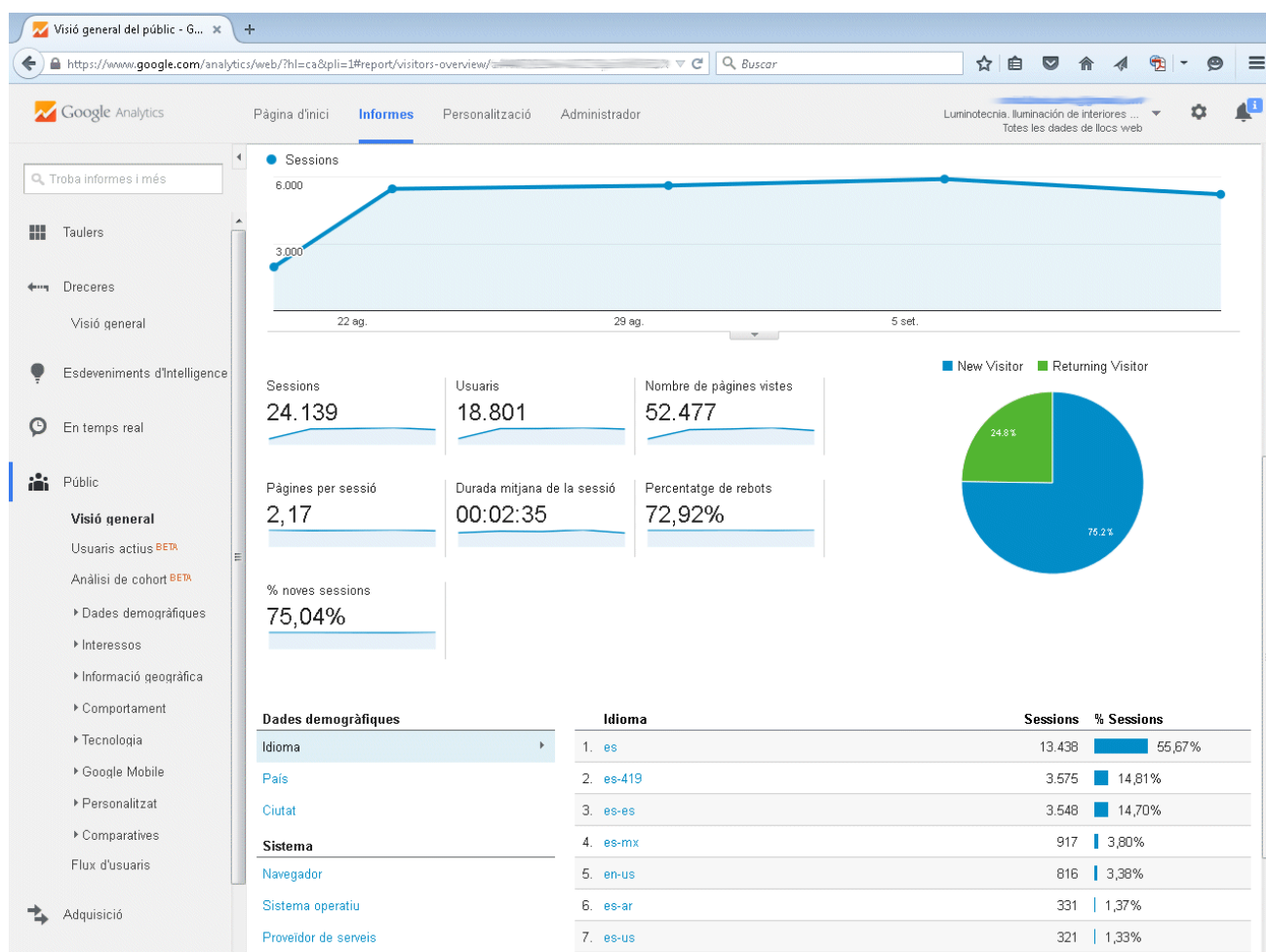


Ilustración 19 Captura de pantalla de Google Analytics del tráfico de usuarios del curso (agosto-septiembre de 2015)

A lo largo de los años se ha ido actualizando, corrigiendo errores y adecuando la tecnología a los cambios del medio. Así se ha pasado de HTML3 en la primera versión a la remozada nueva versión desarrollada en HTML5, CSS3, AJAX con un diseño de la interfaz accesible y que tiene en cuenta los criterios de SEO. Se ha cambiado el sistema de presentación de las fórmulas matemáticas y se ha mejorado la interactividad.

6.1 Planteamiento

Cuando se planificó el curso, se decidió que uno de los objetivos prioritarios fuera que se pudiera seguir en cualquier orden, o por lo menos que no hiciera falta consultar muy profundamente otros capítulos para su seguimiento. No se

¹⁰⁹ Basta con hacer una búsqueda en Google introduciendo edison.upc.edu/curs/llum para encontrar una larga lista de enlaces que apuntan al curso. https://www.google.es/search?q=edison.upc.edu/curs/llum/indice.html&ie=utf-8&oe=utf-8&gws_rd=cr&ei=XLHLVd3jC8WBabOggtgF

pretendía dar una imagen de curso progresivo sino más bien crear una herramienta de autoformación y consulta a la vez.

En esta línea, se decidió que era conveniente evitar los índices numéricos en los capítulos para no dar la sensación de que existía un orden predeterminado. Lo más razonable, parecía pues, dividir el material docente en diferentes unidades temáticas independientes a nivel de contenido y temática, aunque sin caer en repeticiones innecesarias ni perder las ventajas de los enlaces de hipertexto. Por ello, entre otros motivos, se redujeron al mínimo los enlaces entre las diferentes unidades.

Los índices pasaron de ser los típicos listados numerados a convertirse en páginas de enlaces acompañados de una breve explicación del contenido siguiendo el orden lógico. De esta manera, el usuario conoce el orden preestablecido del curso sin verse obligado a cumplirlo. Las páginas de explicaciones no se conciben como una sucesión lógica y ordenada de las mismas, sino como unidades autoconclusivas. Se combinan así, en un equilibrio delicado, autonomía y carácter global.

También es posible navegar avanzando y retrocediendo a través del orden preestablecido utilizando los enlaces generados por la plataforma de publicación como si pasáramos las páginas de un libro.

Otro de los puntos a valorar era el nivel de conocimientos necesario para seguir el curso y el nivel que se iba a ofrecer a los usuarios. En primer lugar, se trataba de hacer textos y explicaciones accesibles a un público de amplio, desde neófitos a profesionales pasando por profesores, estudiantes de ingeniería, etc. Para ello, se pensó que era necesario hacer explicaciones breves, sencillas, con abundantes imágenes y elementos multimedia, etc. Aquellos usuarios que desearan ampliar conocimientos, podrían hacerlo con enlaces desde los que se profundizaría en la explicación de algunos conceptos. Así se consigue un cierto grado de adaptación del contenido a los conocimientos del usuario.

Los resultados obtenidos son muy satisfactorios, a pesar de las dificultades originadas al conjugar objetivos tan dispares. Se ha conseguido crear un curso comprensible para cualquier usuario con conocimientos mínimos de ciencias. En los textos y explicaciones se ha buscado que prevaleciera un carácter divulgativo, progresivo, ameno y sencillo que no fuera demasiado complicado para el lector neófito en la materia ni resultara demasiado elemental para los conocedores del tema. Por este motivo, las explicaciones se han simplificado al máximo, evitando incidir demasiado en temas colaterales como el color, la luz, la visión, la fotometría u otros que, aunque también son importantes, no es necesario conocer muy a fondo para entender la mecánica de los cálculos luminotécnicos. Para otros temas que debido a su gran complejidad quedan fuera del objetivo del curso, como los túneles o la iluminación con proyectores, se ha preferido dar una visión general sin entrar demasiado en detalles. En los ejercicios se han simplificado las exposiciones teóricas y se han resuelto los problemas siguiendo el esquema: lectura del enunciado - interpretación - planteamiento y fórmulas necesarias - solución numérica.

Para aquellos usuarios que desconozcan algunos de los conceptos, pueden acceder, pulsando un enlace, a páginas donde se explican con más detalle, o se les ofrecen demostraciones matemáticas, etc. Se evita así que aquellos usuarios que ya conozcan su significado o no les interese, tengan que seguir estas explicaciones.

6.2 Estructura del contenido del curso

El estudio de la bibliografía sobre iluminación reveló que todos los materiales consultados giraban siempre sobre los mismos temas tratados, eso sí, desde diferentes puntos de vista. Unos ponían más énfasis en el tema de las lámparas, otros en la iluminación de exteriores, etc. El análisis llevó a la conclusión de que el curso se podía dividir en siete unidades temáticas: la luz, la visión, fotometría, lámparas, luminarias, iluminación de interiores e iluminación de exteriores. Se decidió que lo más adecuado era agrupar dos de los temas, lámparas y luminarias, ya que la finalidad del curso era explicar cómo se realizan los proyectos de iluminación, y no ofrecer listados exhaustivos de lámparas, luminarias y sus características. De esta manera el curso se redujo a seis grandes áreas.

- La visión
- La luz
- Fotometría
- Lámparas y luminarias
- Iluminación de interiores
- Iluminación de exteriores

A estas áreas se añadieron complementos, que no forman propiamente parte del curso y están dedicados a ofrecer otras informaciones de interés al usuario como son la bibliografía, los enlaces de interés, la información sobre el proyecto, la licencia de uso y la tabla de contenidos.

6.2.1 La visión

Trata, de manera introductoria, sobre los aspectos fisiológicos de la visión y la percepción visual humanas.

Esta unidad consta de tres páginas. La primera página trata sobre la fisiología del ojo, la segunda sobre la percepción visual y sus características (formación de la imagen, sensibilidad del ojo a la luz, etc.) y la tercera sobre qué factores influyen sobre la visión (tamaño, agudeza visual, etc.). Especialmente significativa es la explicación dedicada a la fisiología. Esta incluye un mapa del ojo interactivo, sobre los nombres del cual el usuario puede pulsar para abrir así ventanas que ofrecen con una explicación de las funciones que realizan. Es una página dónde el usuario participa activamente determinando en todo momento qué conceptos desea conocer.

6.2.2 La luz

Este bloque es una introducción sobre los principios físicos de la luz.

La primera página versa sobre la naturaleza y las principales propiedades de la luz y pretende ser una introducción/refresco de conceptos de física elemental. En él se trata la luz desde el punto de vista físico.

La segunda trata del color. Se puede pensar que el color es una propiedad física de la luz, pero hay que tener en cuenta que también es un fenómeno sensorial y por tanto depende de aspectos fisiológicos y psíquicos. En este punto, se trata la luz desde el punto de vista físico, sensorial, y también se habla de las mezclas.

6.2.3 Fotometría

En el apartado dedicado a la fotometría se explican las magnitudes y unidades empleadas en la luminotecnia y cómo se realizan los cálculos que se utilizan en la práctica mediante el uso de diagramas, gráficos, etc. Además, se incluye una página de ejercicios con ejemplos sobre el tema.

La primera de las tres páginas que forman esta unidad trata sobre las unidades que se utilizan en iluminación. Su contenido es eminentemente teórico, con enlaces con los que el alumno puede ampliar algunos conceptos (el ángulo sólido, la superficie aparente, etc.) y desarrollos de leyes (la ley inversa de los cuadrados o la del coseno). Al final de cada magnitud tratada, se incluye una tabla con su fórmula, su símbolo y las unidades destacadas para facilitar las consultas posteriores.

La segunda desarrolla la aplicación práctica de las magnitudes mediante tablas, gráficos, diagramas, etc. Es una página de carácter técnico, pues en ella se explican las herramientas tradicionales que se emplean en los cálculos luminotécnicos. Se hace especial hincapié en los gráficos, pues esta fue la principal herramienta utilizada para hacer los cálculos hasta la aparición de los programas de cálculo de instalaciones que se emplean actualmente. La comprensión

de los sistemas de cálculo manuales que emplean gráficos y son la base de los algoritmos de cálculo computacional es importante para un correcto uso e interpretación de los resultados de los programas informáticos.

La última página ofrece ejercicios autocorrectivos y ejemplos sobre los puntos tratados en las dos páginas anteriores. La estructura de los ejemplos es la siguiente: empiezan con el enunciado, al que sigue una breve explicación de los fundamentos teóricos y fórmulas necesarias para solucionarlo, finalmente se realizan los cálculos con los resultados. Al final se añade una tabla de resumen con los resultados.

Los problemas propuestos siguen un esquema similar a los ejemplos. En primer lugar, se expone el enunciado al usuario y se le ofrecen dos posibilidades: ver una tabla con los resultados numéricos o ver la solución completa. De esta manera el usuario decide qué prefiere hacer, escogiendo entre probar a resolverlos él mismo o ver cómo se han resuelto.

6.2.4 Lámparas y luminarias

Este tema trata sobre las fuentes de luz artificiales, qué tipos y cómo funcionan, y las lámparas que son los aparatos en los que se instalan.

Las lámparas se dividen temáticamente en lámparas incandescentes y lámparas de descarga (fluorescentes y similares). En cada una se explican los principios físicos y tecnológicos de su funcionamiento y se comentan los tipos que hay, sus ventajas, sus inconvenientes, etc. Como el contenido del curso se desarrolló antes de la aparición en el mercado de las lámparas de LED, estas no se incluyen en el temario.

En las luminarias se describen los aparatos destinados a alojar las lámparas y se comenta cómo son, los tipos que hay, clasificación, diversos aspectos tecnológicos sobre ellas, etc.

6.2.5 Iluminación de interiores

La iluminación de interiores trata sobre los proyectos de iluminación en espacios cerrados. Sin considerar las cuestiones estéticas, se exponen los fundamentos teóricos, las consideraciones técnicas y cómo se realizan los cálculos.

Este tema se divide en tres páginas. En la primera se explica qué aspectos hay que tener en cuenta a la hora de realizar un proyecto de este tipo para garantizar el máximo confort y utilidad a los usuarios. En la segunda se tratan los métodos de cálculo, tanto los simplificados como los numéricos que sirven de base a los programas informáticos. Finalmente, se incluye una página de ejercicios para que los usuarios puedan poner en práctica los conocimientos adquiridos.

6.2.6 Iluminación de exteriores

La iluminación de exteriores ofrece una visión global de la iluminación de espacios públicos a través de un doble punto de vista: el estético y el técnico. A nivel estético se dan recomendaciones generales y algunas indicaciones sobre cómo proceder. Hay que tener en cuenta que cada caso se ha de tratar de forma individual, por lo cual es imposible incluir criterios uniformes ya que la iluminación de exteriores tiene un alto componente de creatividad y esta no se puede enseñar. En el plano técnico, se ofrecen explicaciones o indicaciones sobre cómo se realizan los cálculos y las normativas o recomendaciones emitidas por los organismos competentes.

Respecto al contenido, el tema se estructura en cuatro áreas: alumbrado de vías públicas, alumbrado de áreas residenciales y peatonales, alumbrado de túneles y alumbrado por proyección. Únicamente se incluyen ejercicios para el alumbrado de vías públicas. En los otros casos se ha desestimado porque la mecánica de los cálculos era muy similar (como en el alumbrado residencial o el de proyección) o porque es un tema muy complejo que queda fuera del ámbito del curso como el alumbrado de túneles.

El alumbrado de vías públicas se divide en tres páginas dada la extensión del tema. En la primera el curso ofrece una exposición teórica de las magnitudes y elementos que hay que tener en consideración (luminancia, iluminancia, pavimentos, coeficientes de uniformidad, lámparas, luminarias, deslumbramiento, etc.), ejemplos y comentarios sobre la disposición de puntos de luz que se adoptan en la realidad y los niveles de iluminancia recomendados por la CIE (Comisión Internacional de Iluminación)¹¹⁰. La segunda trata sobre los métodos de cálculo de instalaciones de iluminación de vías públicas poniendo especial énfasis en los métodos de cálculo de iluminancias (método del factor de utilización y métodos numéricos) por encima de los de cálculo de luminancias que requieren de herramientas informáticas de cálculo. Por último, se incluye una página de ejercicios con ejemplos y problemas propuestos para mejor comprensión del usuario del tema.

El alumbrado de áreas residenciales y peatonales ofrece al alumno una breve introducción de un tema carácter multidisciplinar en el que intervienen diseñadores, urbanistas, arquitectos e ingenieros que unen sus experiencias en la creación de ambientes nocturnos. En él se mencionan los requerimientos de este tipo de vías, los niveles de alumbrado necesarios y las clases de lámparas y luminarias que se pueden emplear para esta finalidad.

El alumbrado de túneles, al igual que el anterior, es una introducción a su problemática dado que es un campo de trabajo muy complejo que es difícil sintetizar en tan poco espacio. En él se dan unos esbozos sobre el problema de la iluminación diurna (problemas, requerimientos de iluminación, etc.), la nocturna, las características de los equipos de alumbrado y el mantenimiento que hay que realizar para mantener en óptimas condiciones las instalaciones.

En las últimas dos páginas se explica el alumbrado por inundación o con proyectores. La primera, se dedica a explicar los proyectores (cómo son, propiedades, clasificación, etc.) y los métodos de cálculo. Y la segunda, a las aplicaciones (iluminación de áreas de trabajo o industriales, de edificios y monumentos, de instalaciones deportivas y alumbrado viario) con recomendaciones sobre niveles de alumbrado, ejemplos de disposiciones, etc.

6.3 Estructura del contenido de las páginas

Para unificar el aspecto de las páginas además de compartir un diseño gráfico común se decidió que compartieran la misma estructura de contenido.

Todas las páginas tienen en común un encabezado con el título de la página.

Le sigue una breve introducción sobre el tema de la página donde aparecen enlaces directos a los diferentes apartados para facilitar un acceso rápido. En ocasiones, también se encuentran enlaces a otras páginas donde se explican conceptos relacionados explicados en capítulos anteriores.

El cuerpo de la explicación está dividido en apartados encabezados por su correspondiente título. En él se sitúan las imágenes, las tablas y la exposición del contenido distribuida por párrafos, cada uno sobre un tema concreto.

¹¹⁰ <http://www.cie.co.at/>

6.4 Mapa web del curso

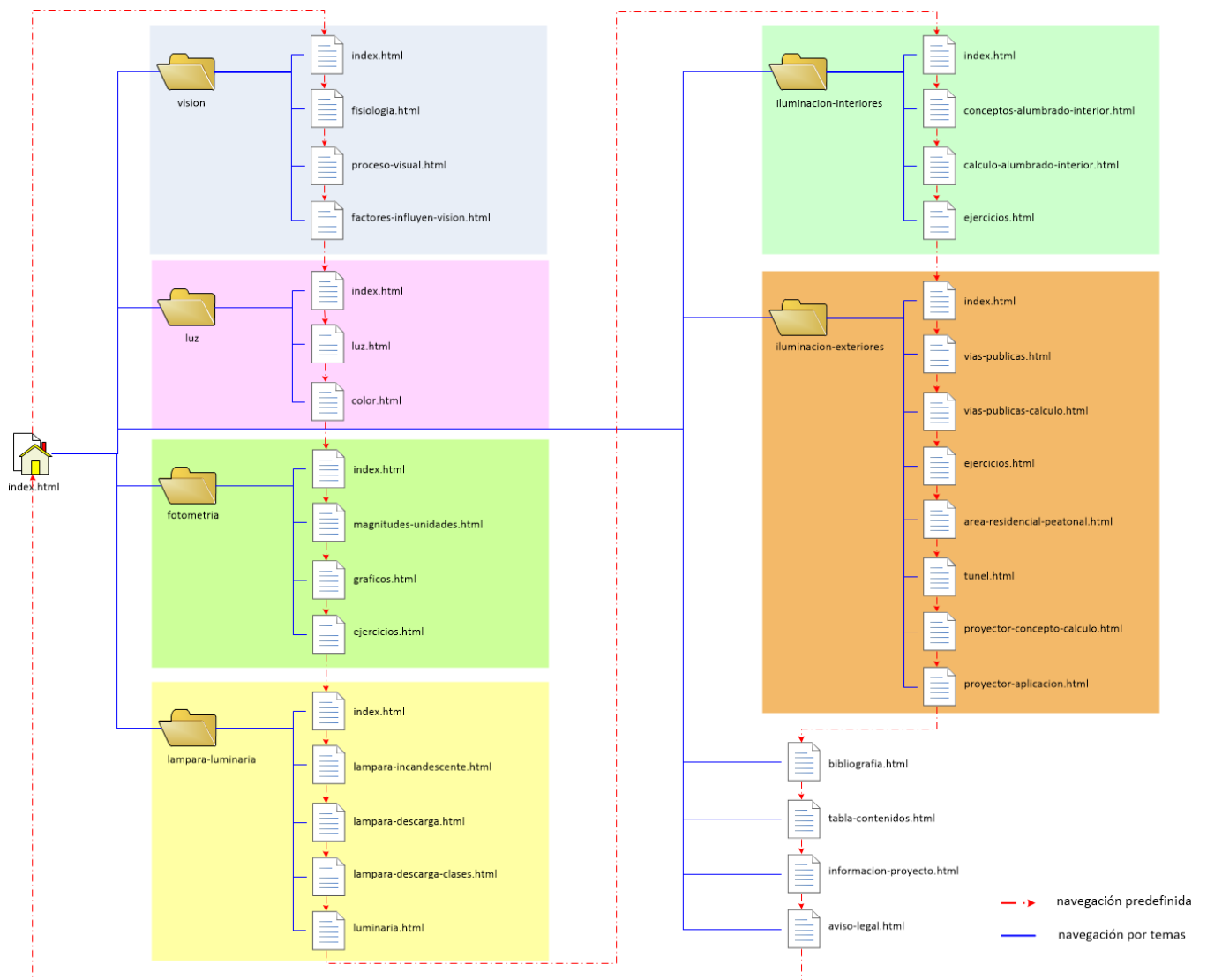


Ilustración 20 Mapa web del curso

7 Publicación de los cursos

Para probar las herramientas del alumno, se optó por crear una plataforma de divulgación de los cursos que simplificara el trabajo de difusión de los mismos. La razón era simple, las herramientas tradicionales de publicación son campus virtuales en los que el docente cuelga documentos, básicamente en formato PDF, enlaces, etc. También se pueden emplear para este fin portales web 2.0, tipo Wikimedia o blogs, donde se introducen los contenidos en HTML pero que no están pensados para publicar cursos hipermedia. Son buenas herramientas, pero no es lo que se buscaba. La idea es que los cursos se asemejen a un libro interactivo.

Como crear cursos utilizando HTML, CSS y javascript puro es complicado para personas sin los conocimientos técnicos adecuados, se optó por:

1. Diseñar una herramienta con tecnología cliente-servidor que a partir de un curso en HTML y un grupo de sencillas instrucciones, diera formato a las páginas, aplicara el estilo adecuado, creara el entorno del curso, generara el sistema de navegación, etc.
2. Crear un conjunto de *scripts* que automatizaran la generación de contenidos multimedia interactivos de forma transparente al usuario.

7.1 Plataforma de publicación

Desde su publicación por primera vez en 1999, el curso de iluminación ha sufrido numerosos cambios. La actualización periódica de sus contenidos demostró que esta tarea llevaba asociados algunos inconvenientes.

El primero de ellos provenía del sistema de navegación, que resultó complicado de mantener. Implementar las modificaciones implicaba reescribir demasiados archivos, con lo que las actualizaciones se convertían en un trabajo farragoso en el que había que invertir muchas horas de trabajo especializado (programación y estilo) que no estaba relacionado con el contenido del curso.

El segundo inconveniente apareció cuando se decidió renovar el aspecto visual de la página. Aunque se utilice CSS, sin los oportunos conocimientos técnicos es un proceso complejo, largo y en el que no siempre se consiguen los resultados esperados.

Por último, cuando se quiere trasladar la experiencia de éste curso a otros, es muy difícil llevarlo a cabo de forma satisfactoria sin unas competencias avanzadas en desarrollo web. Para un formador, lo importante es redactar los contenidos, seleccionar las imágenes y pensar los elementos multimedia que pueden incorporarse al curso. La maquetación y programación no debería ser una limitación a la hora de exponer los conocimientos.

La idea es crear una herramienta que, partiendo de páginas HTML y un conjunto de parámetros de configuración, crea la infraestructura necesaria para la publicación del curso.¹¹¹

7.1.1 Objetivo

Ofrecer un método sencillo y práctico de producir y publicar manuales o cursos de temática científico-técnica. Para conseguir este objetivo sería necesario disponer de los siguientes componentes:

- El *software* para maquetar las páginas web según los parámetros de configuración del curso a partir de los archivos HTML con el texto.

¹¹¹ <http://recursos.citcea.upc.edu/illum/>

- El *software* para administrar los cursos y sus contenidos. De esta manera los responsables de los cursos dispondrán de una herramienta amigable para configurarlos y gestionarlos de forma más sencilla que no editar archivos de configuración.

Cualquiera de las dos partes mencionadas supone una cantidad considerable de trabajo de desarrollo. Dado que, además, había que ocuparse del contenido y maquetación del curso y del desarrollo de las herramientas del estudiante, se optó por centrarse en implementar el primero de los puntos expuestos, que es para poder publicar el curso, y dejar el otro para desarrollos futuros.

Por lo tanto, se ha implementado una plataforma básica (el motor de funcionamiento) junto a un procedimiento manual de instalación y configuración para publicar los cursos.

7.1.2 Funcionamiento

La plataforma de publicación funciona como un cliente web ligero en el que el trabajo de generar la página web que visualiza el usuario se realiza en el servidor. En la siguiente ilustración se puede ver un esquema del funcionamiento del proceso de creación de la página.

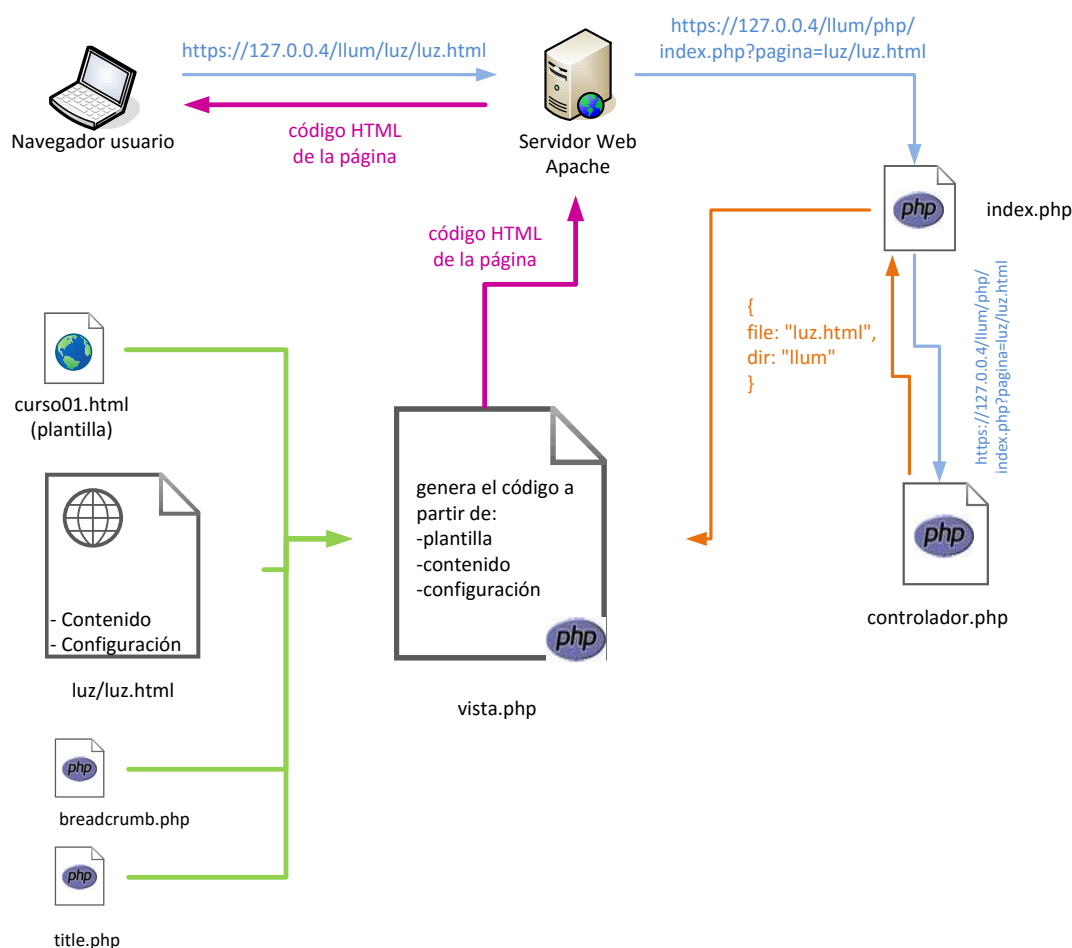


Ilustración 21 Mecanismo de publicación de las páginas del curso en la plataforma

El proceso comienza con el usuario accediendo desde su navegador a una de las páginas del curso. Esto genera la petición HTTP al servidor web de la página, por ejemplo, `https://127.0.0.4/llum/luz/luz.html`¹¹². El servidor web analiza

¹¹² La dirección IP 127.0.0.4 es de pruebas y es un sinónimo de *localhost* (la propia máquina).

la dirección web solicitada, y siguiendo las instrucciones del archivo de configuración local *.htaccess*¹¹³ cambia la página solicitada por <https://127.0.0.4/llum/php/index.php?pagina=luz/luz.html>. Obrando así se consiguen varias cosas:

- Que los enlaces de la página resulten más amigables para las personas, para los robots de los buscadores (SEO) y para las tecnologías de soporte de accesibilidad.
- Que sea más fácil de implementar la arquitectura MVC sobre la que funciona el programa.
- Se oculta la dirección real del programa PHP y se mejora la seguridad del servidor.

El archivo **index.php** se convierte así, en el punto de arranque del programa PHP que genera la página web solicitada. Actúa como correa transmisora de órdenes entre los diferentes elementos de la arquitectura MVC. **index.php** llama al controlador que comprueba que existe la página solicitada y devuelve un objeto con el nombre del archivo y la carpeta dónde encontrarlo.

A continuación, **index.php** invoca la vista que carga en memoria la estructura DOM (modelo de objetos del documento) de la plantilla HTML. La plantilla incluye la información básica de la cabecera (etiquetas meta, CSS, archivos de javascript comunes, iconos...), los contenedores para los diferentes elementos de la página (barras de navegación, artículo principal y pie de página) y el menú de navegación.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title> | Manual de iluminación y luminotecnia</title>
<meta name="keywords" content="luminotecnia, iluminación, interior, exterior, tipos de lámparas,
luminaria, manual" />
<meta name="description" content="Curso de introducción a la iluminación de interiores y
exteriores. Recurso disponible en la web del Departament d'Enginyeria Elèctrica de la UPC
(Barcelona). http://recursos.citcea.upc.edu/llum/" />
<meta name="author" content="Javier García Fernández" />
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<link rel="icon" href="/imatges/bombeta.gif" type="image/gif" />
<link rel="shortcut icon" href="/imatges/bombeta.ico" type="application/x-icon" />
<link rel="stylesheet" type="text/css" href="/css/reset.css" media="all" />
<link rel="stylesheet" type="text/css" href="/css/estilBasic.css" media="screen" />
<script type="text/javascript"> var isIE7 = false; </script>
<!--[if IE 7]>
<script type="text/javascript">isIE7 = true;</script>
<![endif]-->
<script type="text/javascript" src="/js/codi.js" defer="defer"></script>
</head>
<body>
<nav id="menu" role="navigation">
<ul id="nav-lv0">
<li tabindex="0"><a rel="index" title="indice" href="/index.html">Índice</a>
...
</li>
</ul>
</nav>
<div id="contenedor">
<article role="main" id="contenedor-article"></article>
<footer>
<a rel="license" title="aviso" href="/aviso-legal.html">Aviso Legal</a> |
<a title="info" href="/informacion-proyecto.html">Información del proyecto</a>
<br />
<address>&copy; <a rel="nofollow" title="correo"
href="mailto:edison.ee@upc.edu?subject=Manual%20de%20iluminaci%C3%B3n">Javier Garcia Fernandez,
Oriol Boix</a></address>
</footer>
</div>
</body>
</html>
```

Código 14 Fragmento del código de la plantilla

¹¹³ Archivo de configuración local de Apache HTTP Server con directivas de configuración y ejecución del servidor que se aplica a la carpeta donde se encuentra y a sus hijas.

Se realiza la misma operación con la página solicitada por el usuario. Al final del documento se incluyen los datos de configuración necesarios para saber si la página tiene fórmulas y hay que cargar MathJax, si es una página de índice (tienen un estilo diferente) y los datos de navegación (página actual, siguiente y anterior).

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>La luz</title>
</head>
<body>
<h1>La luz</h1>
<p>La luz, que llega a nuestros ojos y nos permite ver, es un pequeño conjunto de radiaciones
electromagnéticas de longitudes de onda comprendidas entre los 380 nm y los 770 nm.</p>

...

<div id="complements"
data-arrowLeft="indixeluz"
data-arrowLeftAddress="index.html"
data-arrowRight="color"
data-arrowRightAddress="color.html"
data-navigation="luz"
data-equations="true"
></div>
</body>
</html>
```

Código 15 Ejemplo de código de una página de contenido con los datos de configuración

Partiendo de la plantilla que se utiliza como documento base, se le aplica un proceso de modificación y combinación con la página HTML con la explicación cuyo resultado final es la página que ve el usuario. Se modifica la cabecera del documento `<head>`; se añade el título que se ha extraído de la página solicitada y los archivos de javascript con MathJax, las herramientas del alumno... Además, se modifican los enlaces para que apunten correctamente y se añaden atributos *title* a los enlaces de navegación y el pie de página para mejorar el SEO, se añaden los paneles de navegación lateral y las "miguitas de pan". Los textos de las migas de pan se extraen del archivo **breadcrumb.php** y los de los atributos *title* que también se utilizan en los paneles laterales de navegación de **title.php**. Finalmente se incorpora dentro de la etiqueta `<article>` el contenido de la página.

La página ya está lista para ser enviada al servidor web y que éste genere la respuesta HTTP, que será transmitida al navegador del cliente.

7.1.3 Programación

En la elaboración de la plataforma se han integrado diferentes tecnologías que hacen posible la publicación del curso. Para implementarlo se ha utilizado una arquitectura MVC¹¹⁴. Es una tecnología de creación de aplicaciones que separa la interacción con el usuario (control), de la gestión de la información (modelo), que en este caso no hay, y de la presentación de la información (vista). Cada uno de ellos se ha implementado utilizando la programación orientada a objetos y el lenguaje PHP.

Un elemento fundamental para que funcione este montaje es que el servidor web transmita al controlador la página web solicitada. En este punto nos encontramos con una disyuntiva. Desde el punto de vista del controlador, lo ideal es que se pase la página web como un parámetro, pero si tenemos en cuenta la accesibilidad y el SEO lo mejor es una dirección amigable.

- Controlador: <https://127.0.0.4/llum/php/index.php?pagina=luz/luz.html>
- SEO, accesibilidad: <https://127.0.0.4/llum/luz/luz.html>

¹¹⁴ Véase el apartado dedicado a MVC

Este problema se resuelve introduciendo un archivo de configuración local de Apache, **.htaccess**, y usando el módulo **rewrite** para realizar las transformaciones en las direcciones.

```
# UTF8
AddDefaultCharset UTF-8
IndexOptions +Charset=UTF-8

<IfModule mod_rewrite.c>
# Redirecciones
RewriteEngine on
RewriteBase /llum/
RewriteCond %{REQUEST_URI} !web-Antigua
# Añadimos index.html al final de las direcciones si no indica el nombre del archivo. Evita
# problemas con los módulos de JS
RewriteRule ^/?$ /llum/index.html [R=301,L]
RewriteRule ^(.*)/$ /llum/$1/index.html [R=301,L]
# Redirección para la arquitectura MVC
RewriteRule ^(.*)\.html$ php/index.php?pagina=$1
</IfModule>
```

Código 16 Archivo **.htaccess**

La línea más importante es la penúltima. En ella, se emplea una expresión regular para capturar la dirección de la página solicitada. Esta, se pasa posteriormente como valor del parámetro "pagina" del archivo **index.php** donde se inicia la ejecución de la aplicación.

El programa, implementado en PHP, se ejecuta en el servidor. PHP es un lenguaje interpretado que se compila en tiempo de ejecución a partir del código fuente que se puede encontrar repartido en diferentes archivos. Así es más fácil y cómodo de mantener. Los archivos que lo integran son:

- index.php
- config.php
- controlador.php
- vista.php
- breadcrumb.php
- title.php

```
<?php
/**
 *
 * @author Javier Garcia Fernandez mc.javiergarcia@gmail.com
 * @copyright (c) 2014, Javier Garcia Fernandez
 * @version 0.1
 *
 * Archivo principal que controla el sistema MVC para generar los cursos
 */
require_once("config.php");
require_once("title.php");
require_once("breadcrumb.php");
require_once("controlador.php");
require_once('vista.php');

try{
    $miControlador = new controlador();
    $miVista = new vista($miControlador->file, $miControlador->dir);
    $miVista->createOutputHTML();
} catch (Exception $ex) {
    echo '<pre>'; var_dump($ex); echo '</pre>';
}
```

Código 17 Archivo **index.php**

index.php es el archivo principal. Cuando se ejecuta, el compilador une su código con el del resto de archivos que aparecen en las instrucciones *require_once()* y monta el programa, lo analiza sintácticamente y lo ejecuta. Este se inicia con la creación de una variable de la clase controlador¹¹⁵. En este proceso, se comprueba que se ha pasado al programa un parámetro llamado "pagina" y que su valor coincide con una de las páginas del curso. A continuación, se crea un objeto de la clase vista y se llama al método *createOutputHTML()* para que monte la página HTML a partir del archivo del curso y la plantilla. En caso de que se produzca algún error de ejecución, se generará una excepción, se interrumpirá la ejecución del programa y se mostrará en pantalla.

```
...

public function __construct($file,$dir) {
    $this->file = $file;
    $this->dir = $dir;
    $this->DOMtemplate = $this->readTemplateHTML();
    $html = $this->readOriginHTML();
    $this->features = $this->readFeatures($html);
    $this->DOMcontents = $this->extractContents($html);
}

...

public function createOutputHTML() {
    $this->editHead();
    $this->editNavigationBar();
    $this->editFooter();
    $this->addArticle();
    $this->addNavigationPanels();
    echo $this->DOMtemplate->saveHTML();
}

...
```

Código 18 Porción del código de la clase vista con el constructor y el método que crea la página de salida

El constructor de la clase vista carga los archivos de la página solicitada y la plantilla (en formato HTML). Utilizando librerías con capacidad para manipular archivos XML, los analiza y almacena cada archivo en un árbol DOM. Aunque trabajar con DOM es complicado, tiene sus ventajas pues permite navegar por los nodos del árbol, sustituir, añadir, eliminar elementos o devolver una cadena HTML con el contenido de una porción de este o el árbol completo. Dentro de la página con el contenido del curso están el texto y los datos de configuración necesarios para montar la página que se envía al usuario.

Antes de optar por esta solución se estudió la posibilidad de trabajar con *strings*. Pero no resultó satisfactoria. En este caso las substituciones implicaban manipular etiquetas HTML además de textos y para ello era necesario utilizar expresiones regulares. Al final la complejidad de estas no compensaba los beneficios aportados ni se observaba un aumento de la velocidad con respecto a la solución adoptada.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>La luz</title>
</head>
<body>
<h1>La luz</h1>
<p>La luz, que llega a nuestros ojos y nos permite ver, es un pequeño conjunto de radiaciones
electromagnéticas de longitudes de onda comprendidas entre los 380 nm y los 770 nm.</p>

<h2>El espectro electromagnético</h2>

<p>La luz forma parte del espectro electromagnético que comprende tipos de ondas tan dispares
como los rayos cósmicos, los rayos gamma, los ultravioletas, los infrarrojos y las ondas de
```

¹¹⁵ POO

radio o televisión entre otros. Cada uno de estos tipos de onda comprende un intervalo definido por una [magnitud característica](#window-01) que puede ser la longitud de onda (λ) o la frecuencia (f). Recordemos que la relación entre ambas es:

```
<figure class="formula">
\[\lambda = \frac{c}{f}\]
<figcaption>Relación entre la frecuencia y la longitud de onda</figcaption>
</figure>
```

...

```
<!-- Inici finestres -->
<div id="window-01" class="extra">
<h2>Comentario</h2>
<p>Estrictamente hablando, la única magnitud característica debería ser la frecuencia ya que la longitud de onda depende de la velocidad de la luz y esta a su vez del medio que atraviesa (vacío, gas, líquido, sólido transparente, etc). Pero como nos referiremos siempre a longitud de onda en el vacío no le daremos más importancia al tema.</p>
</div>
```

...

```
<!-- Fi finestres -->

<div id="complements"
data-arrowLeft="indicaluz"
data-arrowLeftAddress="index.html"
data-arrowRight="color"
data-arrowRightAddress="color.html"
data-navigation="luz"
data-equations="true"
></div>
</body>
</html>
```

Código 19 Fragmento de la página con el texto antes de ser procesada

Una vez extraído el texto y las opciones de configuración, se inicia el proceso de montaje final de la página con la llamada a la función *createOutputHTML()*. Esta crea o modifica los diferentes elementos de la página, la cabecera, el pie de página, los paneles de navegación, el texto y se ajusta las URL de enlaces, imágenes y archivos para que todo funcione correctamente.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>La luz | Manual de iluminación y luminotecnia</title>
<meta name="keywords" content="luminotecnia, iluminación, interior, exterior, tipos de lámparas, luminaria, manual">
<meta name="description" content="Curso de introducción a la iluminación de interiores y exteriores. Recurso disponible en la web del Departament d'Enginyeria Elèctrica de la UPC (Barcelona). http://recursos.citcea.upc.edu/llum/">
<meta name="author" content="Javier Garcia Fernandez">
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<link rel="icon" href="/llum/imatges/bombeta.gif" type="image/gif">
<link rel="shortcut icon" href="/llum/imatges/bombeta.ico" type="application/x-icon">
<link rel="stylesheet" type="text/css" href="/llum/css/reset.css" media="all">
<link rel="stylesheet" type="text/css" href="/llum/css/estilBasic.css" media="screen">
<script type="text/javascript"> var isIE7 = false; </script>
<!--[if IE 7]>
<script type="text/javascript">isIE7 = true;</script>
<![endif]-->
<script type="text/javascript" src="/llum/js/codi.js" defer></script>
<script type="text/x-mathjax-config">MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$'], ['\\(','\\)']],displayAlign: "left"}});</script>
<script type="text/javascript"
src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML"></script>
</head>
<body>
<nav id="menu" role="navigation">
```

```

<ul id="nav-lv0"><li tabindex="0"><a rel="index" title="Índice"
href="/llum/index.html">Índice</a>
    &gt; <a href="index.html">La luz y el color</a> &gt; La luz
    <ul id="nav-lv1"><li><a title="Teoría del funcionamiento de la visión humana"
href="/llum/vision/index.html">La visión</a>
    ...
</ul>
</nav>
<div id="contenedor">
<article role="main" id="contenedor-article">
<h1>La luz</h1>
<p>La luz, que llega a nuestros ojos y nos permite ver, es un pequeño conjunto de radiaciones
electromagnéticas de longitudes de onda comprendidas entre los 380 nm y los 770 nm.</p>

<h2>El espectro electromagnético</h2>

<p>La luz forma parte del espectro electromagnético que comprende tipos de ondas tan dispares
como los rayos cósmicos, los rayos gamma, los ultravioletas, los infrarrojos y las ondas de
radio o televisión entre otros. Cada uno de estos tipos de onda comprende un intervalo definido
por una <a rel="nofollow" href="#window-01">magnitud característica</a> que puede ser la
longitud de onda (?) o la frecuencia (f). Recordemos que la relación entre ambas es:</p>

...

<!-- Inici finestres -->
<div id="window-01" class="extra">
<h2>Comentario</h2>
<p>Estrictamente hablando, la única magnitud característica debería ser la frecuencia ya que la
longitud de onda depende de la velocidad de la luz y esta a su vez del medio que atraviesa
(vacío, gas, líquido, sólido transparente, etc). Pero como nos referiremos siempre a longitud de
onda en el vacío no le daremos más importancia al tema.</p>
</div>

...

<!-- Fi finestres -->

</article>
<footer>
<a rel="license" title="Aviso legal, licencia de uso y derechos de autor" href="/llum/aviso-
legal.html">Aviso Legal</a> |
<a title="Información sobre los objetivos del manual, tecnologías empleadas y novedades"
href="/llum/informacion-proyecto.html">Información del proyecto</a>
<br><address>© <a rel="nofollow" title="Correo electrónico de contacto con los autores"
href="mailto:edison.ee@upc.edu?subject=Manual%20de%20iluminaci%C3%B3n">Javier Garcia Fernandez,
Oriol Boix</a></address>
</footer></div>

<nav id="arrow-left">
<div id="arrow-left-symbol"><a rel="start" title="Teoría de la luz y el color"
href="index.html">&lt;</a></div>
<div id="arrow-left-text"><a rel="start" title="Teoría de la luz y el color" href="index.html">
Teoría de la luz y el color</a></div>
</nav>

<nav id="arrow-right">
<div id="arrow-right-symbol"><a rel="next" title="¿Qué es el color y como es percibido?"
href="color.html">&gt;</a></div>
<div id="arrow-right-text"><a rel="start" title="¿Qué es el color y como es percibido?"
href="color.html">
¿Qué es el color y como es percibido?</a></div>
</nav>
</body>
</html>

```

Código 20 Fragmento de la página después de ser procesada

El código HTML resultante, generado por PHP, se envía al servidor web que a su vez, lo reenvía al navegador del usuario.

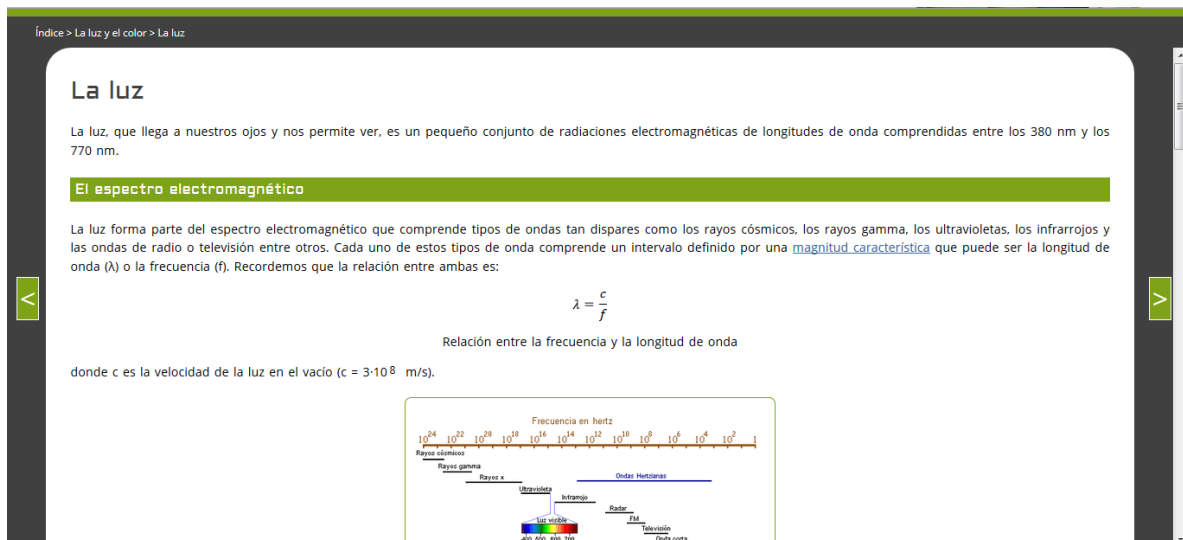


Ilustración 22 Aspecto final de la página montada

7.1.4 Estructura de carpetas del programa

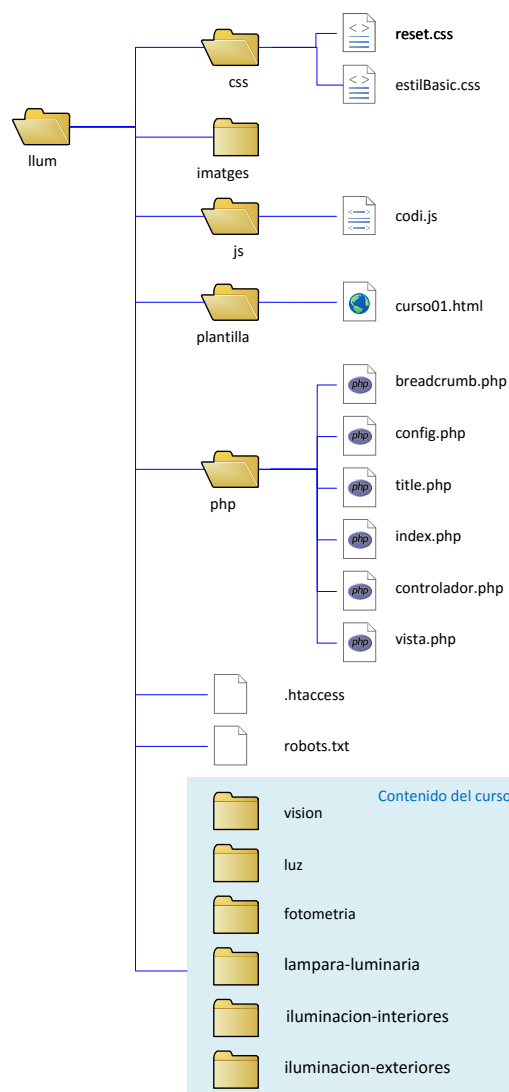


Ilustración 23 Estructura de carpetas de la plataforma de publicación

7.1.5 Uso

Como ya se ha mencionado, esta herramienta se ha creado para dar únicamente soporte al curso de demostración. Por tanto no se ha realizado un trabajo muy profundo con ella y su instalación y configuración es de carácter manual.

La instalación se hace en la misma carpeta del curso HTML. En ella hay que copiar los archivos **.htaccess** y **robots.txt** y las carpetas php, imatges, plantilla., css y js. A continuación hay que editar los siguientes archivos:

- **.htaccess**, cambiar la carpeta base del curso
- **robots.txt**
- En **config.php** modificar la ruta del curso y la de las herramientas si fuera necesario
- Definir las rutas de las miguitas de pan en **breadcrumb.php**
- Definir los textos de los enlaces para los atributos *title* y los paneles de navegación lateral
- Modificar el menú de navegación de la plantilla y opcionalmente su aspecto
- Cambiar la hoja de estilo CSS (opcional)
- Editar todas las páginas de los cursos y añadirles una etiqueta `<div>` con los datos de configuración

Los datos de configuración de cada página han de añadirse siguiendo el siguiente formato:

```
<div id="complements"
data-arrowLeft="indixeluz"
data-arrowLeftAddress="index.html"
data-arrowRight="color"
data-arrowRightAddress="color.html"
data-navigation="luz"
data-index="true"
data-equations="true"
></div>
```

Código 21 Configuración de la página

Los valores de *data-arrowXXX* sirven para construir los paneles de navegación lateral que permiten avanzar y retroceder en el curso. Cada uno de ellos coincide con una entrada de un *array* indexado¹¹⁶ del archivo **title.php**. *data-navigation* determina la ruta de la página que seguirán las migas de pan y coincide con el índice del *array* indexado de **breadcrumb.php**. El penúltimo parámetro sirve para definir si la página es un índice (si es así, se le aplicará un estilo diferente). El último parámetro indica si hay que incluir los archivos de MathJax. Estos dos parámetros sólo se ponen si son necesarios.

7.1.6 Desarrollos futuros

En un futuro, sería deseable continuar evolucionando la plataforma de publicación para hacer su uso más sencillo y amigable. Se convertiría así, en un gestor de contenidos CMS con soporte de usuarios y herramientas para la publicación y edición de los contenidos. Entre las funciones relacionadas con la publicación de contenidos a desarrollar de forma prioritaria estarían:

- Subir las páginas al servidor como archivos HTML
- Generar la tabla de contenidos (TOC) que serviría para:
 - Establecer el orden para pasar las páginas. De esta manera se podrían generar automáticamente los paneles de navegación
 - Crear los menús de navegación del curso
 - Crear las migas de pan

¹¹⁶ Son aquellos vectores en que los índices son textos en lugar de números. En otros lenguajes de programación se denominan diccionarios.

- Gestionar las plantillas de publicación de los cursos
- Gestionar las hojas de estilo de los cursos

Además de todo esto, también sería interesante incorporar un editor para crear las páginas web al estilo de Wikimedia¹¹⁷.

7.2 Scripts para los contenidos multimedia

Para hacer más sencilla y rica la publicación de los contenidos del curso, se han añadido dos herramientas que se ejecutan en el navegador del usuario (cliente).

La primera de ellas sirve para incluir fórmulas científicas en las páginas HTML.

La segunda permite incorporar pequeñas aplicaciones interactivas en las páginas de forma sencilla y transparente a personas sin conocimientos de programación.

7.2.1 Fórmulas matemáticas en documentos hipermedia

Desde el comienzo de la WWW la presentación de símbolos y fórmulas matemáticas en documentos científicos ha sido un quebradero de cabeza.

La solución más fácil y universal, para que fuera visible en todos los navegadores, es incluir la fórmula como una imagen en el documento HTML. Esta se puede crear directamente con un programa de tratamiento de imágenes, o con un editor de texto con capacidad para escribir fórmulas y exportarla como una imagen. El problema viene cuando hay que modificarla. Imaginemos que hay que cambiar el tamaño de la fuente; hay que abrir el archivo original, realizar los cambios, volver a generar la imagen y subirla al servidor. Dado que la imagen habrá cambiado de tamaño, se deberá modificar sus medidas en la instrucción HTML correspondiente. Un proceso largo, tedioso y laborioso sobre todo cuando se trabaja en textos científicos con abundancia de fórmulas.

La solución podría ser escribir la fórmula como un texto normal. Así, la edición resultaría mucho más fácil o, en su defecto, escribirla con alguna notación textual que pudiera modificarse de forma sencilla. En el caso de símbolos sueltos o fórmulas simples -que no incluya sumatorios, divisiones, raíces...- se pueden utilizar actualmente entidades HTML o escribir directamente los caracteres codificándolos en UTF8.

$$\alpha = \beta - \theta \cdot t$$

Código 22 Ejemplo de escritura de fórmulas matemáticas con UTF8

```
&alpha; = &beta; - &#952; &middot; t
α = β - θ · t
```

Código 23 Ejemplo de escritura de fórmulas matemáticas con entidades HTML

Sin embargo, cuando la fórmula es compleja, este método no es válido y es necesario algún tipo de sintaxis que permita escribir ecuaciones como MathML.

MathML¹¹⁸ es un lenguaje creado por el W3C para representar fórmulas matemáticas. Es un lenguaje de etiquetas derivado de XML y forma parte de las recomendaciones de HTML5. El navegador se encarga de leer el documento y

¹¹⁷ <https://www.mediawiki.org/wiki/MediaWiki>

¹¹⁸ <http://www.w3.org/Math/>

transformarlo en una imagen. La sintaxis es complicada en comparación a la notación anterior como puede verse en el siguiente ejemplo.

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mi mathvariant="normal">α</mi>
  <mo>=</mo>
  <mi mathvariant="normal">β</mi>
  <mo>-</mo>
  <mi mathvariant="normal">θ</mi>
  <mi>·</mi>
  <mi mathvariant="normal">t</mi>
</math>
```

$$\alpha = \beta - \theta \cdot t$$

Código 24 Ejemplo de código MathML

Sin embargo, es más adecuada para escribir fórmulas complejas.

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <msub>
    <mi mathvariant="normal">Φ</mi>
    <mi>E</mi>
  </msub>
  <mo stretchy="false">=</mo>
  <mrow>
    <msub>
      <mo stretchy="false" largeop="true">∮</mo>
      <mi>S</mi>
    </msub>
    <mover accent="true">
      <mi>E</mi>
      <mo stretchy="false">·</mo>
    </mover>
    <mo stretchy="false">d</mo>
    <mi>d</mi>
    <mover accent="true">
      <mi>s</mi>
      <mo stretchy="false">·</mo>
    </mover>
  </mrow>
</math>
```

$$\Phi_E = \oint_S \vec{E} \cdot d\vec{s}$$

Código 25 Fórmula del flujo del campo eléctrico escrita en MathML

Aunque parezca que ya se ha encontrado una solución, no es así. Por diferentes motivos¹¹⁹, algunos navegadores ofrecen un soporte limitado o directamente no implementan MathML. Por ejemplo, los navegadores que utilizan los motores de *renderizado*¹²⁰ o *engine layout* Blink y Webkit, entre los que se incluyen Chrome (uno de los navegadores más populares), Opera y Apple Safari y las versiones antiguas de Internet Explorer. Esto lo convierte en una solución inviable.

Debido a los problemas de falta de soporte y la necesidad de simplificar la notación, se han ideado soluciones en el cliente y en el servidor que generan la imagen a partir de una notación de texto¹²¹.

¹¹⁹ https://en.wikipedia.org/wiki/MathML#Web_browsers [84]

¹²⁰ Componente de *software* del navegador encargado de convertir el código fuente de la página (HTML, CSS, imágenes...) en la página que visualiza el usuario. Los principales motores que existen en la actualidad son: Blink (Chrome, Opera), Gecko (Firefox), Trident (Internet Explorer) y Webkit (Android browser y Apple Safari).

¹²¹ <https://developer.mozilla.org/en-US/docs/Web/MathML/Authoring> [126]

La notación de texto, *simple text o linear syntax* consiste en escribir una fórmula empleando únicamente caracteres de texto, generalmente ASCII. Conviene destacar que estas notaciones se utilizan como medio para escribir ecuaciones, pero en sí mismas no sirven para generar la imagen; de esto se encarga una librería. Actualmente las dos soluciones más populares son:

- TeX/LaTeX.
- AsciiMath.

TeX es una herramienta tipográfica especialmente pensada para crear documentos científicos y técnicos (artículos, libros, revistas...) que incorporen fórmulas matemáticas complejas. Es muy popular en ambientes académicos, en especial la versión LaTeX que añade macros para facilitar la composición de los documentos. Por ejemplo, Wikipedia lo utiliza para la escritura y visualización de fórmulas. En TeX, las órdenes comienzan por `\` y los argumentos van entre llaves `{}`. Su ámbito va más allá de utilizarse simplemente para escribir fórmulas matemáticas.

```
E_{H}=\frac{I\cdot\cos^{3}\alpha}{d^{2}}
```

$$E_H = \frac{I \cdot \cos^3 \alpha}{d^2}$$

Código 26 Ejemplo de fórmula escrita en LaTeX

AsciiMath es un sistema de notación pensado especialmente para mostrar expresiones matemáticas en navegadores web. Su desarrollo es reciente y se utiliza en combinación con librerías de javascript que generan la visualización. Las fórmulas van entre ``` y la sintaxis puede consultarse en su página web¹²².

```
`E_H=(I·cos^3 alpha)/d^2`
```

$$E_H = \frac{I \cdot \cos^3 \alpha}{d^2}$$

Código 27 Ejemplo de fórmula en notación AsciiMath

Una herramienta que genere las fórmulas en el servidor funciona siguiendo los siguiente principios. Se sustituye en la página web la dirección URL de la imagen con la ecuación por una llamada al programa que la genera. Como parámetro se pasa la fórmula en alguna notación de texto tal como se ve en el siguiente ejemplo.

```

```

$$E_H = \frac{I \cdot \cos^3 \alpha}{d^2}$$

Código 28 Ejemplo de uso de una solución de servidor con LaTeX para generar fórmulas matemáticas

En el servidor se recibe la petición, se ejecuta el programa y se crea la imagen que se envía al cliente. Para aumentar la velocidad de ejecución, se puede añadir una caché de archivos con las imágenes de las fórmulas que se vayan generando; así se evita volver a ejecutar el programa cada vez.

En versiones previas del proyecto, se desarrolló un programa generador de imágenes a partir de fórmulas escritas en LaTeX con PHP. El algoritmo del programa es el siguiente:

1. PHP lee la fórmula escrita en LaTeX
2. Genera una clave única md5 que identifica la fórmula
3. Comprueba la existencia de la fórmula en la caché de imágenes

¹²² <http://asciimath.org/>

4. Si está, se pasa al punto 8; en caso contrario continúa el proceso
5. Genera un archivo de texto con la fórmula en formato LaTeX
6. El compilador de LaTeX genera un archivo compilado DVI
7. Otro programa convierte el DVI en PNG
8. Se envía la imagen PNG al navegador del cliente

Esta solución tiene la ventaja de que funciona siempre y en cualquier navegador, al contrario de las soluciones de cliente basadas en javascript que dependen de que esté activado. Las soluciones en el servidor implican instalar programas adicionales que conviertan la fórmula en imagen y esto no siempre es posible cuando la web se aloja en un *hosting*¹²³. También hay que tener en cuenta que la imagen no es indexada por los robots de búsqueda. Por todo ello, hubo de ser descartada.

Por el contrario, las soluciones basadas en cliente funcionan con una combinación de javascript, HTML y CSS. Son fáciles de utilizar; se añade un enlace para cargar un archivo a la página y se ejecutan en el navegador del usuario. Pero tienen el problema de que no funcionan si el usuario tiene desactivado javascript, los robots de búsqueda no indexan las fórmulas, el proceso es lento pues las formulas se han de generar cada vez que se carga la página y pueden haber conflictos con el código HTML o con otros javascripts. Aun así, a día de hoy, es la mejor solución por ser la que ofrece máxima compatibilidad y sencillez de uso.

Existen muchas librerías para generar fórmulas como MathJax¹²⁴, ASCIIMathML¹²⁵, LaTeXMathML¹²⁶ o jqMath¹²⁷. En este proyecto se ha optado por la primera porque es la más popular y potente de todas ellas.

MathJax es una librería *cross-browser* escrita en javascript que muestra expresiones matemáticas en navegadores web. Admite fórmulas escritas en notación MathML, TeX o AsciiMath y es capaz de generar visualizaciones en formato HTML+CSS, MathML y SVG. Su licencia de uso es *Open Source* y está patrocinada por importantes instituciones como la Sociedad Americana de Matemáticas, la SIAM, IEEE... [46] [47]

```
<figure class="formula">
\left[E_H\right]=\frac{I\cdot\cos^3\alpha}{h^2}
<figcaption>Componente horizontal de la ley del coseno</figcaption>
</figure>
```

$$E_H = \frac{I \cdot \cos^3 \alpha}{h^2}$$

Componente horizontal de la ley del coseno

Código 29 Ejemplo de uso de MathJax con notación en LaTeX

7.2.2 Complementos educativos

Con el objetivo de mejorar las capacidades interactivas del curso, se han desarrollado varias aplicaciones en javascript. Aunque cuando se hizo el curso original en 1998, suponía un avance respecto a lo que era habitual en Internet¹²⁸, a día de hoy ya existen muchas librerías que ofrecen utilidades similares como jQuery, AngularJS¹²⁹, etc. Como uno de los

¹²³ Empresa que se dedica a ofrecer servicios de alojamiento para páginas web.

¹²⁴ <https://www.mathjax.org/>

¹²⁵ <http://asciimath.org/>

¹²⁶ <http://math.etsu.edu/LaTeXMathML/>

¹²⁷ <http://mathscribe.com/author/jqmath.html>

¹²⁸ En esa época, lo habitual era que las aplicaciones interactivas de las páginas web se programaran en Flash y *ActionScript*.

¹²⁹ <https://angularjs.org/>

criterios del proyecto era evitar el uso de librerías de terceros y ninguna de las probadas resultaba satisfactoria; se optó por rehacer los antiguos *scripts* adaptándolos a los nuevos estándares y paradigmas de programación.

Dos de los criterios más importantes que se han tenido en cuenta para el proyecto eran que las aplicaciones fueran accesibles siguiendo las normas WCAG y ARIA, y que siguieran las recomendaciones de SEO. Como ya se ha comentado, estos dos objetivos acostumbran a ir de la mano.

Era importante conseguir que el contenido de las aplicaciones fuera visible aunque javascript estuviera desactivado. Esto es útil para los robots de indexación, que ignoran el código de javascript mientras analizan la página, y para la accesibilidad, puesto que no se oculta contenido. Además, se ha cuidado que los enlaces y elementos interactivos sean visibles y se pueda navegar por ellos con el teclado.

Otro elemento importante que se aplicó en el proyecto es utilizar código no intrusivo. No se utilizan definiciones de CSS dentro de las páginas ni aparecen controladores de eventos en las etiquetas de enlaces <a> que se han definido en un archivo separado.

Las aplicaciones se han empaquetado dentro de un único módulo. El patrón modular de programación en javascript permite crear un entorno de ejecución del código aislado que es independiente del resto de código que existe en la página. De esta manera se evitan conflictos con otros programas que podrían provocar efectos imprevisibles. Es una técnica muy utilizada y recomendable.

```
var myMódul = (function () {
    var proPrivada1, propPrivada2...;
    ...
    function functionPrivada1() { ... };
    function functionPrivada2() { ... };
    ...
    return {
        proPublica1: proPrivada1,
        functionPublica1: functionPrivada1
    }
})();
```

Código 30 Estructura de un módulo en javascript

Javascript es un lenguaje funcional orientado a objetos en el que el ámbito de ejecución¹³⁰ de las variables y métodos es la función donde están definidas; al contrario de otros lenguajes como Java, C... en que es el bloque de código donde está definido (una estructura condicional, iterativa, función...). Esto permite que cualquier código incluido dentro de una función sea "propiedad" de esta. Si le añadimos que javascript permite asignar una función a una variable, obtenemos la posibilidad de definir de forma sencilla un módulo (algo equivalente a un espacio de nombres). La estructura *return*, que sirve para especificar que devuelve una función, permite convertir en públicos y accesibles desde el exterior del módulo métodos y propiedades que son privados a este.

Para utilizar las aplicaciones hay que añadir una referencia al archivo con el script en la cabecera de la página; pero no es necesario que el editor lo haga puesto que la plataforma de publicación ya se encarga de ello.

```
<script type="text/javascript" src="/llum/js/codi.js" defer=""></script>
```

Código 31 Código HTML que hay añadir a la cabecera del documento

El uso de las aplicaciones es transparente al usuario, no necesita saber programar para utilizarlas. Como se detallará más adelante, basta con incluir un sencillo código HTML. Por ejemplo, en el caso de las ventanas emergentes¹³¹ sólo hay

¹³⁰ El ámbito de ejecución es la porción del código en el que una variable o función está definido y se puede trabajar con él.

¹³¹ Ventana que se abre en primer plano.

que incluir su contenido dentro de una etiqueta <div>. El diseño no es intrusivo. Cuando se carga la página, se ejecuta un programa que busca las etiquetas predefinidas, y mediante una manipulación del código HTML utilizando javascript, DOM y CSS se ocultan contenidos, se crean formularios de control, etc. Además, se han definido controladores de eventos para que se ejecuten ciertas acciones como respuesta a acciones del usuario (pulsar un botón, seleccionar un elemento de una lista...) que no son objeto de posterior programación, por lo que el creador del curso no ha de actuar sobre ellos.

Las aplicaciones diseñadas son las siguientes:

- Un generador de ventanas emergentes o *pop-up*
- Un sistema para seleccionar la imagen a mostrar de una lista para los ejercicios
- Un sistema para esconder/mostrar selectivamente porciones de la página HTML
- Un sistema de explicaciones interactivas que va cambiando texto, una tabla e imágenes secuencialmente al pulsar unos botones
- Una tabla interactiva de colores que cambia el color de la celda cuando se pasa el ratón por encima

7.2.2.1 Ventanas emergentes

Uno de los propósitos definidos al inicio del proyecto era poder ofrecer diferentes niveles de conocimientos para que el curso se adaptara a las necesidades de los usuarios. La idea es que los alumnos que lo desearan pudieran profundizar más en ciertos temas sin que ello rompiera el hilo conductor de la explicación. Así se podía prescindir de las demostraciones matemáticas de las fórmulas o de incluir cuadros explicativos con definiciones de conceptos que no formaban parte del argumento principal. El acceso a las explicaciones avanzadas y/o complementarias se hace a través de enlaces que abren ventanas emergentes en la pantalla del navegador. Los enlaces de este tipo se diferencian del resto en que cuando se pasa el ratón por encima se envuelven con un icono que recuerda a una ventana.



Ilustración 24 Ventana emergente

Cuando el usuario pulsa el enlace, el programa coloca un fondo semitransparente sobre la ventana del navegador, determina automáticamente sus medidas, calcula el tamaño necesario de la ventana emergente y la muestra centrada en pantalla. Todo esto se hace de una forma transparente para el desarrollador de contenidos.

El uso de la aplicación es muy sencillo como puede verse en el siguiente código:

```

1. HTML con el contenido de la ventana al final del documento

<div id="window-06" class="extra">
<h2>Iris y pupila</h2>
<p>El iris está situado detrás de la córnea y delante del cristalino con una abertura en el
centro llamada pupila cuya función es regular la cantidad de luz que entra en el ojo; abriéndose
en condiciones de oscuridad y cerrándose si la intensidad de luz es elevada.</p>
</div>

2. HTML para abrir la ventana

<a rel="nofollow" href="#window-06" title="Iris y pupila">Iris y pupila</a>

```

Código 32 HTML para crear y usar ventanas emergentes

En primer lugar se escribe el código con el contenido de las ventanas, introduciéndolo dentro de una etiqueta <div> con la forma <div id="window-XX" class="extra"> donde XX es el número de la ventana. No es necesario incluir ningún código para cerrar la ventana ya que el programa crea el enlace con el texto "CERRAR" automáticamente. Los enlaces para abrir las ventanas tienen la forma de un enlace a un marcador de página, empiezan por #, seguidos del nombre de la ventana "window-XX".

Si se observa el código fuente de la página HTML se ve que cumple con los criterios de accesibilidad de WCAG y con las recomendaciones de SEO. Cuando javascript está desactivado, el contenido de las ventanas está visible al final de la página; y se puede acceder a ellas a través de los enlaces de marcador definidos.

Para simplificar el uso de esta aplicación, se ha pensado que en futuras versiones se podrían utilizar etiquetas personalizadas que el programa se encargaría de leer y transformar en etiquetas normativas como se muestra a continuación.

```

HTML con el contenido de la ventana

<popupWindow number="XX">...</popupWindow> -> notación alternativa
<div id="window-XX" class="extra">...</div> -> notación normativa

HTML para abrir la ventana

<a rel="nofollow" href="#popupWindow-06" title="Iris y pupila">Iris y pupila</a>

```

Código 33 Idea para simplificar la notación en el caso de las ventanas emergentes

7.2.2.2 Selección interactiva de imágenes

Cuando se planteó la inclusión de ejercicios en el curso para ayudar a la asimilación de conceptos, surgió una complicación relacionada con la presentación gráfica de éstos. En iluminación, antes de la aparición de los actuales programas de cálculo, era habitual trabajar con gráficos y tablas e interpolar datos para obtener los valores necesarios para hacer los cálculos. Esto suponía que cada ejercicio tenía que incluir un número variable de imágenes (dos, tres, cuatro o más) que alargaban la página y le daban un aspecto lineal, como si se tratara de un libro.

8. Para el tramo de calle de la figura, calcular la iluminancia en los puntos a, b, c, d, e y f. La farola mide 8 m de altura y la lámpara tiene un flujo de 15000 lm. Asimismo, se suministran los diagramas polares de las luminarias referenciadas a 1000 lm.

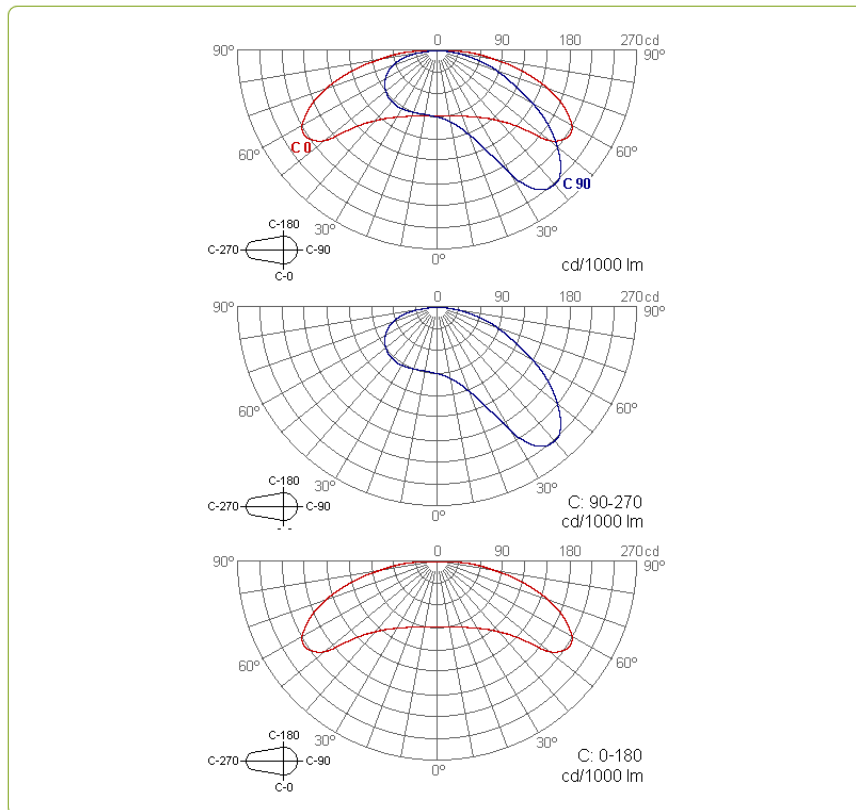
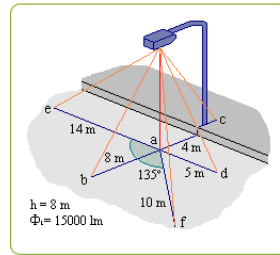
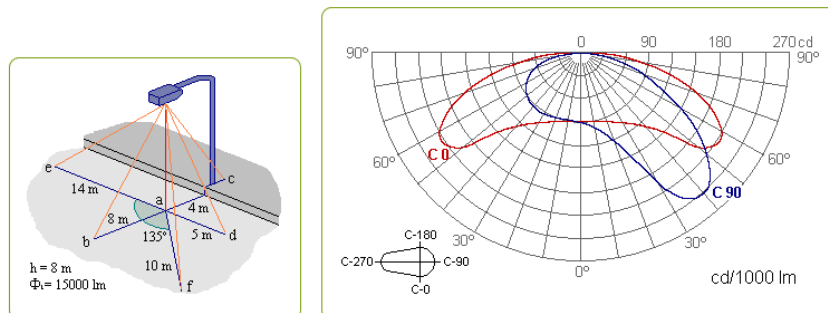


Ilustración 25 Ejemplo del enunciado de un ejercicio sin javascript

Como la idea era crear un manual interactivo, se buscó mejorar la presentación de los ejercicios y hacerlos más amigables para el usuario. Así se reducía la cantidad de información en pantalla y resultaban menos confusos. Por ello se creó una aplicación a través de la cual es posible seleccionar la imagen que se quiere ver de una lista desplegable o mediante unos botones.

Las listas desplegables se han utilizado en los enunciados de los ejercicios para ofrecer la información de forma más compacta.

8. Para el tramo de calle de la figura, calcular la iluminancia en los puntos a, b, c, d, e y f. La farola mide 8 m de altura y la lámpara tiene un flujo de 15000 lm. Asimismo, se suministran los diagramas polares de las luminarias referenciadas a 1000 lm.



Diagramas polares disponibles:

Ambos
 Ambos
 Longitudinal C: 0-180
 Transversal C: 90-270

[Ver resultados](#)

Ilustración 26 Aspecto del enunciado con la imagen seleccionada de una lista desplegable

El uso de la aplicación se muestra a continuación:

```
<div class="figure-container">
<figure></figure>
<figure>
<div id="exercici-canvi-imatge-figura-0" data-label="Selecciona un diagrama">



</div>
</figure>
<div class="figcaption">Diagramas polares</div>
</div>
```

Código 34 HTML para la selección de una imagen con una lista desplegable

El código final resulta muy simple y compacto. La lista de imágenes debe estar dentro de una etiqueta <div> con la forma <div id="exercici-canvi-imatge-figura-X" data-label="Selecciona un diagrama"> donde X es un número correlativo por si hubiera más de un ejercicio de este tipo en la página y *data-label* el texto de la etiqueta que acompaña a la lista. El texto de las opciones de la lista se pone en el atributo *alt* que acompaña a las imágenes.

Los botones para seleccionar la imagen se han utilizado para reforzar la explicación de un ejercicio resuelto. Cada vez que se pulsa un botón el gráfico se sitúa sobre uno de los puntos marcados en el plano. El código es similar al del caso anterior.

Una vez realizado esto, podemos pasar a superponer la gráfica sobre las diferentes luminarias y leer los valores de la curva isolux sobre los puntos tal y como hicimos al hablar del [método de los nueve puntos](#).

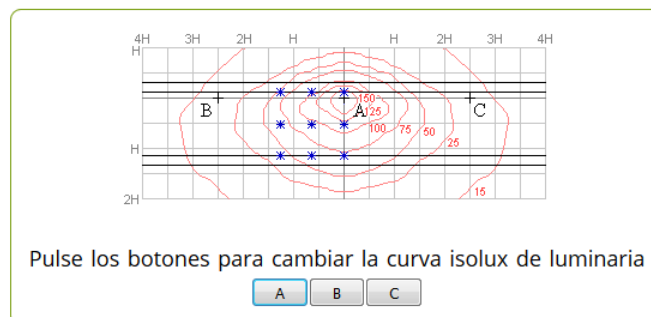


Ilustración 27 Botones de selección de la imagen en una explicación

```
<div class="figure-container">
<figure>
<div id="exercici-canvi-button-imatge-figura-0" data-label="Pulse los botones para cambiar la
curva isolux de luminaria" >



</div>
</figure>
</div>
```

Código 35 HTML para la selección de una imagen con botones

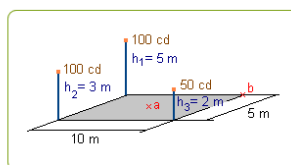
Los criterios para programar la aplicación son los mismos que en el caso de las ventanas emergentes. Se ha buscado que sean visibles aún con javascript desactivado y se ha añadido código ARIA necesario para ayudar a las tecnologías de soporte.

7.2.2.3 Mostrar/ocultar selectivamente el contenido de una página

Además de los ejercicios resueltos con los que se inicia cada página de ejercicios, existen también una serie de ejercicios propuestos para que el alumno pueda practicar y afianzar sus conocimientos.

En los ejercicios propuestos inicialmente, el usuario solo puede ver el enunciado y los datos necesarios para resolverlo. Pero si lo considera necesario, dispone de dos enlaces que le permiten visualizar contenidos ocultos: la tabla abreviada con los resultados y la solución comentada del problema.

7. Para la disposición de luminarias de la figura, calcular la iluminancia en el centro de la placa (a) y en el punto b.



[Ver resultados](#)

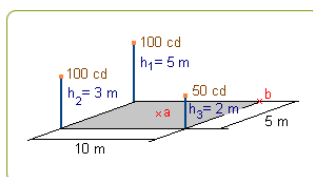
[Ver solución](#)

Ilustración 28 Ejemplo de ejercicio propuesto

La razón de actuar de esta manera es didáctica. Si la solución estuviera inicialmente visible es muy probable que los usuarios se sintieran tentados de mirarla. De esta manera, la consulta se convierte en un acto voluntario, con lo que aumenta la probabilidad de que el alumno intente primero resolver la cuestión y después comprobar que su

planteamiento ha sido el correcto, comprobando sus resultados mirando la tabla correspondiente. Si no logra resolverlo puede consultar la solución completa con todos los pasos para resolver el problema.

7. Para la disposición de luminarias de la figura, calcular la iluminancia en el centro de la placa (a) y en el punto b.

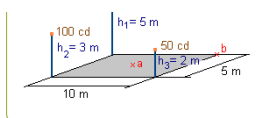


[Ver resultados](#)

Punto	E (lux)
a	2.84
b	1.19

Resultados

Ilustración 29 Ejercicio con resultados desplegados



[Ver resultados](#)

[Ver solución](#)

Solución

En este caso la diferencia radica en que hay más de una fuente de luz, pero esto no ha de suponer una mayor dificultad. Como las iluminancias sobre un punto son aditivas, lo que hay que hacer es calcular la contribución de cada foco sobre dicho punto y sumarlas.

$$E_p = \sum_{i=1}^n \frac{I_i \cdot \cos^3 \alpha_i}{h_i^2} = E_{1p} + E_{2p} + E_{3p}$$

donde



Ilustración 30 Solución del ejercicio

La experiencia dice que, a veces, cuando no se sabe solucionar un problema, lo mejor es ver el planteamiento, comprenderlo y terminar por resolverlo uno mismo. De esta manera, la próxima vez que se plantee un problema similar será más fácil asociarlo a los problemas estudiados y conseguir resolverlo.

Al igual que las aplicaciones anteriores, su uso es muy sencillo.

```

<p><a rel="nofollow" href="#resultado-07">Ver resultados</a></p>

<div id="resultado-07" class="exercicis">
<table>
<caption>Resultados</caption>
<thead>
<tr><th>Punto</th><th>E (lux)</th></tr>
</thead>
<tbody>
<tr><td>a</td><td>2.84</td></tr>
<tr><td>b</td><td>1.19</td></tr>
</tbody>
</table>
</div>

<p><a rel="nofollow" href="#solucion-07">Ver solución</a></p>

<div id="solucion-07" class="exercicis">
<h3>Solución</h3>

<p>En este caso la diferencia radica en que hay más de una fuente de luz, pero esto no ha de
suponer una mayor dificultad. Como las iluminancias sobre un punto son aditivas, lo que hay que
hacer es
calcular la contribución de cada foco sobre dicho punto y sumarlas.</p>

...

</div>

```

Código 36 Ejemplo de uso de la aplicación que muestra/oculta contenidos de forma selectiva

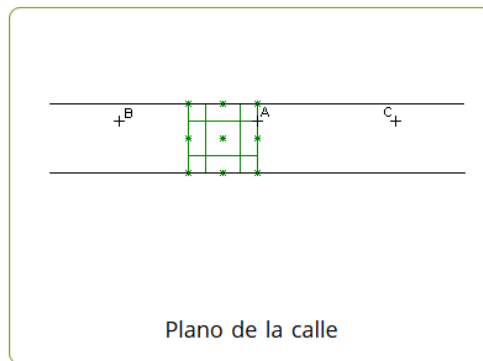
La idea es similar a la de las ventanas emergentes. El control de la aplicación se realiza con un enlace normal; de esta manera no es necesario definir CSS extra ni javascript para que se pueda navegar por él con el teclado. Apparentemente es un enlace a un marcador interno de la página que comienza con #solucion-XX o #resultado-XX donde XX es un número correlativo. El texto del marcador coincide con el del atributo id de la etiqueta <div> donde se encuentra el contenido que debe ocultarse o mostrarse.

7.2.2.4 Explicaciones interactivas

La idea de esta aplicación parte de cómo trasladar la explicación paso a paso del método de los nueve puntos¹³² a un medio interactivo utilizando un ejemplo numérico. Hacerlo de forma lineal ocuparía mucho espacio en la página y gráficamente no sería demasiado atractivo porque supondría mostrar una larga sucesión de imágenes y tablas. Como las características de la explicación se asemejan a una secuencia animada, se pensó que la mejor opción era crear algo similar a un tutorial con botones para avanzar en la explicación, retroceder y reiniciar la secuencia.

Como se puede observar a continuación, cada imagen corresponde a un paso de la resolución del problema. La interfaz de usuario está formada por tres elementos y los botones de navegación. En la parte superior hay una imagen donde se enseña cómo colocar y leer un diagrama isolux. En la inferior una tabla donde se van anotando los valores leídos del gráfico y los cálculos realizados. Y en el medio se sitúa un texto con la explicación de cada paso realizado.

¹³² Es un método numérico de cálculo que se emplea para calcular las iluminancias sobre una serie de puntos representativos de la superficie que se quiere iluminar. Es la base de los métodos utilizados en los programas de cálculo.



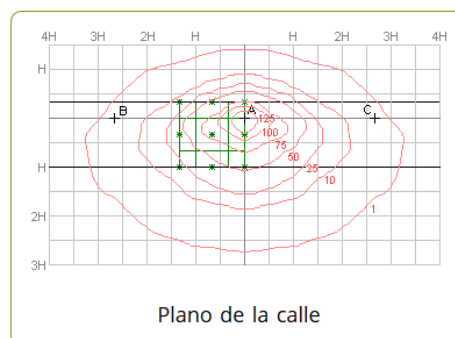
	1	2	3	4	5	6	7	8	9
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
ΣE_{ic}	0	0	0	0	0	0	0	0	0
$E_{i \text{ real}}$	0	0	0	0	0	0	0	0	0

Tabla de recogida de datos y cálculo de resultados

Atras Inicio Adelante

Ilustración 31 Paso 1 de la explicación interactiva

Se empieza con un diagrama de la calle y la tabla para recoger los datos vacía.



Sobre el plano de la calle superponemos la curva isolux sobre una de las luminarias.

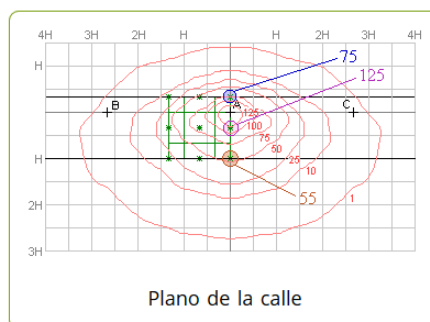
	1	2	3	4	5	6	7	8	9
A	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
ΣE_{ic}	0	0	0	0	0	0	0	0	0
$E_{i \text{ real}}$	0	0	0	0	0	0	0	0	0

Tabla de recogida de datos y cálculo de resultados

Atras Inicio Adelante

Ilustración 32 Paso 2 de la explicación interactiva

En el segundo paso, se muestra una imagen con la curva superpuesta sobre el plano de la calle.



A continuación leemos los valores relativos de la iluminancia en cada punto y los anotamos en la tabla.

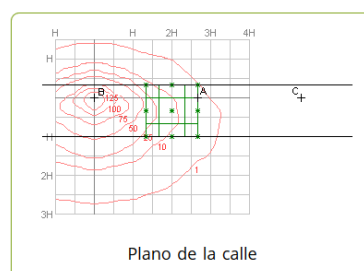
	1	2	3	4	5	6	7	8	9
A	75	125	55	50	80	45	15	45	20
B	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
$\Sigma E_{i\ c}$	0	0	0	0	0	0	0	0	0
$E_{i\ real}$	0	0	0	0	0	0	0	0	0

Tabla de recogida de datos y cálculo de resultados

[Atras](#) [Inicio](#) [Adelante](#)

Ilustración 33 Paso 3 de la explicación interactiva

En el tercero se trasladan las lecturas del plano a la tabla.



Una vez terminado, trasladamos la curva isolux a otra luminaria y repetimos el proceso.

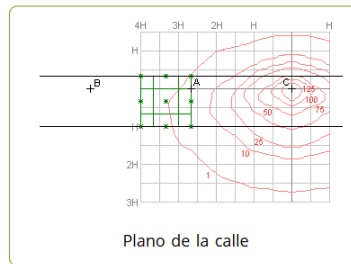
	1	2	3	4	5	6	7	8	9
A	75	125	55	50	80	45	15	45	20
B	1	5	4	7	15	8	18	45	20
C	0	0	0	0	0	0	0	0	0
$\Sigma E_{i\ c}$	0	0	0	0	0	0	0	0	0
$E_{i\ real}$	0	0	0	0	0	0	0	0	0

Tabla de recogida de datos y cálculo de resultados

[Atras](#) [Inicio](#) [Adelante](#)

Ilustración 34 Paso 4 de la explicación interactiva

Ahora se muestra un gráfico donde se cambia de punto la curva isolux y se trasladan las nuevas lecturas a la tabla.



Finalmente, sumamos las contribuciones individuales de cada luminaria sobre cada uno de los puntos, y obtenemos sus iluminancias relativas. Por último, sólo queda calcular los valores reales aplicando la fórmula.

	1	2	3	4	5	6	7	8	9
A	75	125	55	50	80	45	15	45	20
B	1	5	4	7	15	8	18	45	20
C	1	5	4	0	1	0	0	0	0
ΣE_{ic}	77	135	63	57	96	53	33	90	40
$E_{i \text{ real}}$	24.1	42.2	19.7	17.8	30	16.6	10.3	28.1	12.5

Tabla de recogida de datos y cálculo de resultados

[Atras](#) [Inicio](#) [Adelante](#)

Ilustración 35 Paso 5 de la explicación interactiva

Finalmente se completa la operación para el punto C, se calculan los totales y los valores reales de la iluminancia

A continuación, puede observarse el código necesario para crear la secuencia. No incluye los botones, que como ya se ha comentado, se generan automáticamente.

```
<div id="explicacio-sequencia-0">
  <div id="explicacio-subsequencia-0">
    ...
  </div>
  <div id="explicacio-subsequencia-1">
    ...
  </div>
  <div id="explicacio-subsequencia-2">
    ...
  </div>
  <div id="explicacio-subsequencia-3">
    ...
  </div>
  <div id="explicacio-subsequencia-4">
    ...
  </div>
</div>
```

Código 37 Ejemplo de uso de explicaciones secuenciales

Es un código similar al empleado en la selección interactiva de imágenes.

7.2.2.5 Tabla interactiva de combinaciones de colores

Con esta aplicación se pretende reforzar el conocimiento sobre el efecto de la luz coloreada sobre los objetos de color; se trata de entender que el color no es una cosa absoluta, sino que depende de la naturaleza del objeto iluminado y de la luz. Para ello se dispuso una tabla en que las filas coinciden con el color de los objetos y las columnas con el color de la luz empleado para iluminarlos. De esta manera, la combinación fila-columna nos da el color percibido. Como poner una tabla de interacciones de color sólo con texto resulta poco didáctico e impropio del medio, se optó porque al pasar el cursor por encima de la celda cambiara el color.

Efecto de la luz coloreada sobre los objetos de color

Color del objeto	Color de la luz			
	Rojo	Amarillo	Verde	Azul
Blanco	Rosa claro	Amarillo muy claro	Verde muy claro	Azul muy claro
Amarillo	Naranja rojizo	Naranja brillante claro	Amarillo verdoso claro	Marrón rojizo claro
Rojo	Rojo brillante	Rojo brillante	Rojo amarillento	Rojo azulado oscuro
Marrón	Rojo marrón	Naranja marrón	Marrón aceitunado oscuro	Marrón azulado
Negro	Negro rojizo	Negro anaranjado	Negro verdoso	Negro azulado
Azul oscuro	Púrpura rojizo oscuro	Púrpura rojizo claro	Azul verdoso oscuro	Azul brillante
Azul claro	Azul rojizo	Azul rojizo oscuro	Azul verdoso	Azul brillante
Verde	Verde aceituna	Verde amarillento	Verde brillante	Azul verdoso

Efecto de la luz coloreada sobre los objetos de color

CERRAR

Ilustración 36 Tabla de colores interactiva

```
<table data-table-color="true">
<caption>Efecto de la luz coloreada sobre los objetos de color</caption>
<thead>
<tr>
<th rowspan="2">Color del objeto</th>
<th colspan="4">Color de la luz</th>
</tr>
<tr>
<td style="background-color:#FF0000"><span class="blanco">Rojo</span></td>
<td style="background-color:#FFFF00">Amarillo</td>
<td style="background-color:#33FF33">Verde</td>
<td style="background-color:#3333FF"><span class="blanco">Azul</span></td>
</tr>
</thead>
<tbody style="background-color:#FFFFFF;">
...
<tr>
<td style="background-color:#66FFFF">Azul claro</td>
<td data-background="#D7A3FE">Azul rojizo</td>
<td data-background="#860099" data-color="#FFFFFF">Azul rojizo oscuro</td>
<td data-background="#02FDC5">Azul verdoso</td>
<td data-background="#4040FF" data-color="#FFFFFF">Azul brillante</td>
</tr>
...
</tbody>
</table>
```

Código 38 Porción de código de la tabla de colores interactiva

El programa detecta automáticamente la presencia de la tabla y registra los eventos necesarios para que las celdas cambien de color. El color de la celda se define con el atributo *data-background* y el color del texto con *data-color* (opcional). Dada la complicación de la aplicación, se ha recurrido al uso de atributos *style* para definir los colores de las celdas del color de la luz y del objeto aunque no es una solución muy elegante.

Ciertamente, no es una aplicación muy atractiva para el alumno. A día de hoy, se diseñaría una interfaz de usuario más interactiva; del estilo de la figura que se presenta a continuación.

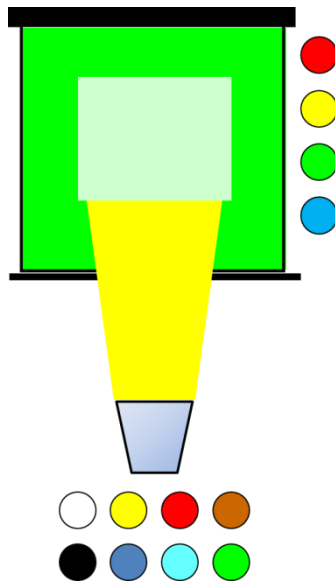


Ilustración 37 Propuesta de nueva interfaz de usuario para la tabla de colores

En ella se combinaría javascript con una imagen SVG y CSS. El alumno podría escoger el color de la luz y el del objeto, en este caso la pantalla; y en un recuadro dentro de esta se vería el color resultante. En la imagen se muestra el caso en que la luz fuera amarilla y la pantalla verde.

8 Herramientas del estudiante

Las herramientas del estudiante o *Web Student Tools* se han creado para hacer más dinámicos, participativos y amigables los cursos en línea estáticos. Hacen posible que el alumno interactúe con los contenidos, pueda modificarlos y hacerlos suyos. Estimulan al usuario en el proceso de aprendizaje animándole a participar y no ser un mero espectador.

Los cursos en formato HTML a menudo adolecen de ser una mera traslación de libros publicados en formato papel a un nuevo medio con algún aditamento extra (a veces, únicamente enlaces). Las herramientas de *Web Student Tools* avanzan en este sentido, ofreciendo al alumno la oportunidad de construir conocimientos "a medida".

Con esta finalidad, se ha buscado que tengan una imagen simple, con un diseño, lo suficientemente intuitivo para que el usuario no necesite manual de instrucciones. Cualquier persona familiarizada con los paquetes de ofimática e internet puede hacerse una idea de cómo funcionan los iconos.

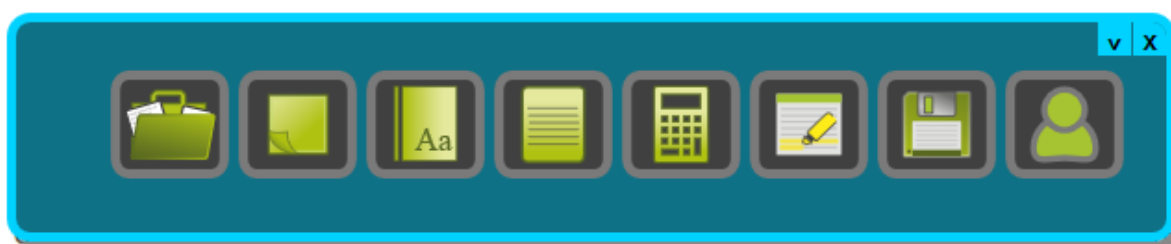


Ilustración 38 Aspecto de la barra de herramientas

Cada usuario dispone de su propio espacio para guardar sus contenidos y poderlos recuperar desde cualquier dispositivo conectado a Internet. Así se aprovechan las capacidades de almacenamiento en la red y la disponibilidad que ofrece este medio.

Las herramientas se han diseñado para que sean autónomas unas de otras, con el mínimo de interacciones posibles entre ellas, excepto, lógicamente, con la barra contenedora. De esta manera se facilita su integración y el desarrollo futuro de nuevas utilidades.

Constituyen, por tanto, el núcleo central en torno al que gira este trabajo y en el que más tiempo se ha invertido en investigación, autoformación e implementación de la misma.

8.1 Arquitectura de la aplicación

Las herramientas del estudiante nacieron de la implementación de un puñado de ideas de cómo hacer más amigable y adaptables los contenidos de los cursos estáticos en línea. Para unificar estas ideas en un proyecto, fue necesario articularlas y empaquetarlas en una barra de herramientas con un sistema de control de usuarios para garantizar que cada persona tuviera el acceso y control de sus propios contenidos. A medida que la complejidad iba en aumento, fue necesario estructurar el código para hacerlo más modular y facilitar así el desarrollo de nuevas herramientas individuales que se pudieran añadir a la aplicación en el futuro.

Fue necesario diseñar una arquitectura, un patrón de implementación para cada nueva herramienta que le permitiera integrarse fácilmente en la aplicación. Aunque la idea básica es simple, en la práctica no lo es tanto. Durante el desarrollo de los primeros módulos surgieron numerosos problemas que, a medida que se fueron resolviendo, hicieron necesario modificar la arquitectura.

Cada herramienta de la barra es un módulo independiente del resto, que funciona de forma autónoma, pero sigue una estructura común que garantiza el correcto funcionamiento e integración en la aplicación. Para este proyecto se implementaron siete herramientas incluyendo la propia barra.

- barra y gestión de usuarios
- calculadora
- notas
- subrayado
- glosario
- resumen
- portafolio

Las herramientas siguen un esquema cliente-servidor pesado¹³³ en el que la generación de la interfaz UI y el control de las acciones del usuario recaen en el navegador del usuario, y la gestión de la información se realiza en el servidor. La vista y el controlador del modelo MVC se implementan en el cliente (navegador) y el modelo en el servidor.

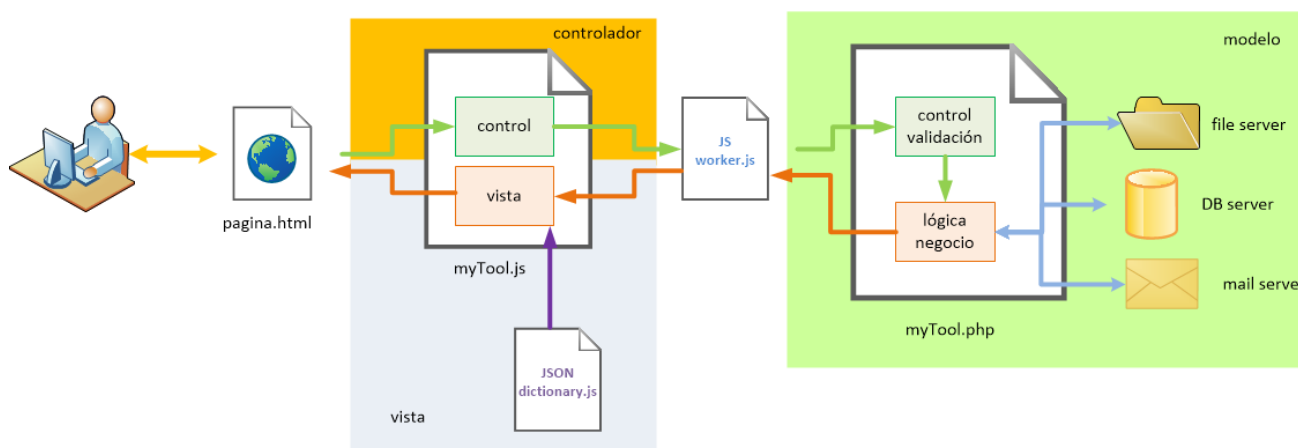


Ilustración 39 Arquitectura básica y funcionamiento de una herramienta

La imagen anterior sirve para ilustrar cómo funciona y se estructura el código. Existen tres archivos principales, dos en el cliente y otro en el servidor, que se encargan de gestionar el funcionamiento de cada herramienta.

En el cliente, todo gira en torno un archivo implementado en javascript, al que nos referiremos como **myTool.js**, que se encarga de crear y manipular la interfaz, gestionar las interacciones con el usuario, enviar peticiones al servidor y tratar la respuesta.

La UI¹³⁴ se implementa con HTML, CSS y javascript. Cuando se carga el módulo, una rutina escrita en javascript crea el código HTML de la interfaz, define todos los controladores¹³⁵ necesarios para gestionar las acciones del usuario y carga los archivos auxiliares con los datos de configuración, idioma y estilo. El aspecto se define en un archivo CSS y los mensajes de texto y la configuración en archivos JSON; de esta manera se consigue que todo sea más sencillo de manipular. La regla de oro es "todo lo que sea susceptible de ser cambiado, configurado, tenga texto en algún idioma, o defina un estilo debe estar un archivo externo". A fin de cuentas, es más fácil cambiar un archivo de texto que buscar el código correspondiente dentro del programa. El fichero **myTool.js** realiza la doble función de actuar como controlador y

¹³³ Un cliente web pesado es aquel en que la carga de generación de la interfaz de usuario se realiza en el navegador del usuario.

¹³⁴ Interfaz de usuario o *User Interface*.

¹³⁵ Un controlador de eventos es una rutina del programa que se mantiene alerta para que cuando se produzca una determinada acción, como pulsar un botón, se ejecute un conjunto de instrucciones.

como vista. Por un lado gestiona la interacción del usuario con la UI y por otro la modifica como respuesta a sus acciones.

El archivo principal del servidor, al que nos referiremos genéricamente como **myTool.php**, se encarga de gestionar la lógica de negocio de la herramienta. Está programado en PHP y es el responsable de interactuar con la base de datos, enviar correo electrónico, manipular ficheros, etc. Cada vez que recibe una petición HTTP del cliente, valida el tipo de acción solicitada (guardar datos en la BD, recuperarlos...) y los datos recogidos, realiza las operaciones pertinentes y devuelve un mensaje de respuesta al cliente. Para ello, recibe el soporte de una librería de funciones encargadas de las tareas de validación, conexión con la base de datos, sistema de archivos, correo electrónico...

La comunicación entre ambos se realiza mediante AJAX y el formato de transmisión de datos utilizado es JSON.

AJAX permite reducir la cantidad de información transmitida entre cliente y servidor. Gracias a esto se consigue un significativo aumento de velocidad y, como consecuencia, una mayor comodidad de uso para el usuario. Durante la implementación del sistema de comunicación surgieron tres graves dificultades relacionadas con la naturaleza asíncrona de javascript y el uso de múltiples hilos de ejecución concurrentes¹³⁶.

Javascript es un lenguaje en el que muchas de sus APIs funcionan de forma asíncrona en lugar de lineal. En un código lineal, las instrucciones se ejecutan una tras otra; hasta que no acaba una acción, no se inicia la siguiente. Este hecho es especialmente relevante si se llama a una función o se accede a recursos externos como BD, archivos, páginas web... en los que el sistema ha de esperar un tiempo hasta recibir la respuesta.

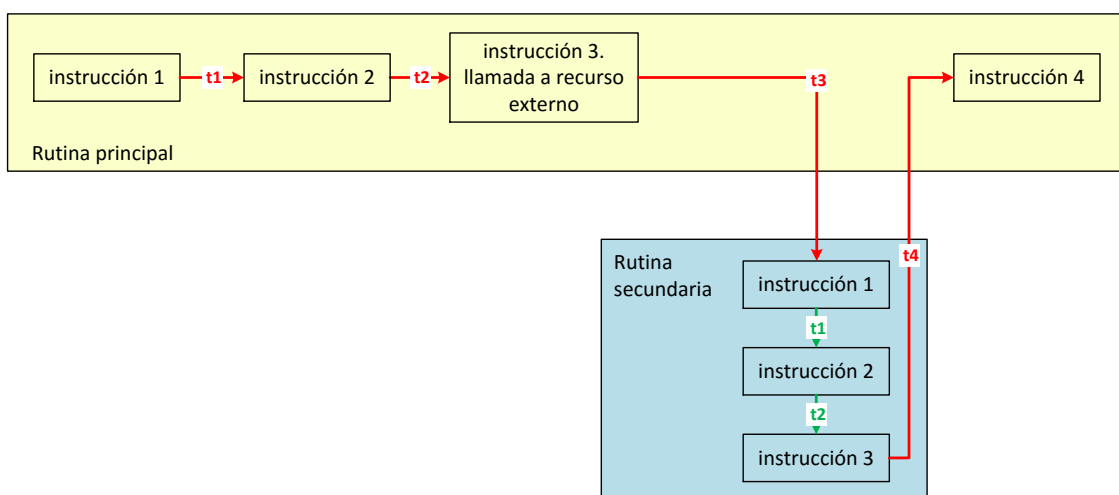


Ilustración 40 Ejecución de una rutina en forma secuencial

En un sistema asíncrono, el programa principal no espera a la respuesta y continúa ejecutándose. Cuando la rutina secundaria termina, envía la respuesta al programa principal donde otra rutina – el controlador – se mantiene a la espera para procesarla. De esta manera, los tiempos de ejecución se reducen y hay menos interrupciones, pero obligan a utilizar un paradigma diferente de programación. Hay que definir controladores para trabajar con las respuestas y tener en cuenta que el flujo de ejecución ha cambiado y que se ejecutarán acciones sin que estén finalizados los pasos previos.

¹³⁶ Procesos que se ejecutan de forma simultánea e independiente.

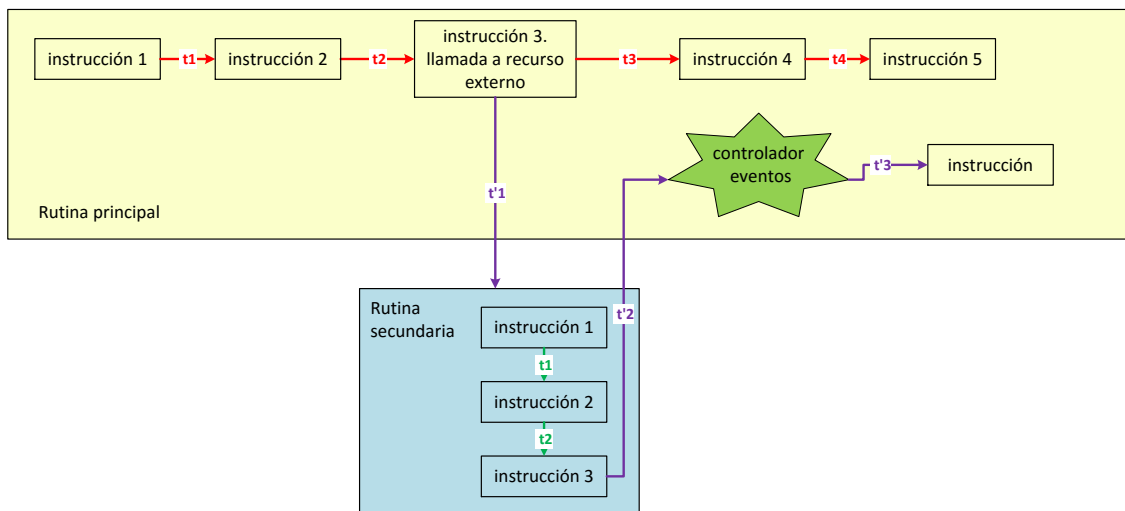


Ilustración 41 Ejecución de una rutina en forma asíncrona

Cuando se trabaja con AJAX, la comunicación con el servidor funciona de forma asíncrona. Por tanto, es necesario definir un controlador de eventos que esté alerta cuando llega la respuesta del servidor. Se desconoce el momento exacto en el que llegará la respuesta, pero sí se sabe que cuando ocurra se ejecutará una determinada rutina.

La primera dificultad provocada por el uso de javascript y AJAX está relacionada con el hecho de que el protocolo HTTP carece de estado. El problema es que, si por ejemplo, hay tres funciones que envían mensajes al servidor, no es posible conocer cuál de ellas hizo la petición. No es complicado de solucionar, se puede incorporar una variable de estado al módulo, solución adoptada, o un parámetro a los mensajes.

La segunda plantea un problema mucho más difícil de solucionar. En una página HTML todas las rutinas de javascript corren en un único hilo de ejecución. Por tanto, si se envía más de una petición simultáneamente resulta que el controlador sólo procesa una única respuesta. Este inconveniente es realmente grave, ya que al haber varios módulos trabajando a la vez, se pierden datos y provoca que sea imposible controlar el flujo de ejecución.

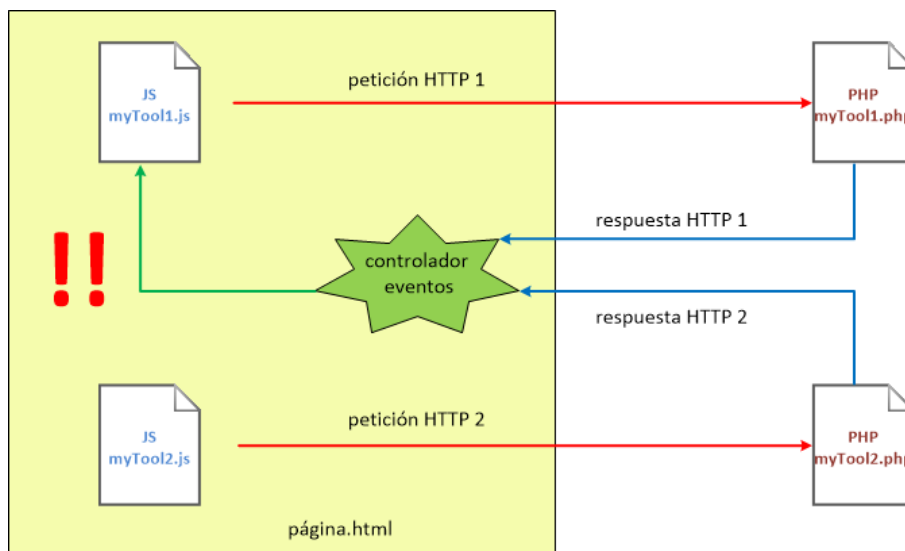


Ilustración 42 El problema de las peticiones concurrentes en AJAX

La solución pasó por aplicar una de las nuevas API de HTML5, *worker*, que permite trabajar con múltiples hilos de ejecución o *threads*. Así, se definió un hilo de trabajo secundario para cada herramienta que controla la comunicación de esta con el servidor. Ahora los procesos están separados y son independientes entre sí.

En javascript, los hilos de ejecución se implementan en un archivo separado, **worker.js**, que intercambia mensajes con el archivo principal de la aplicación.

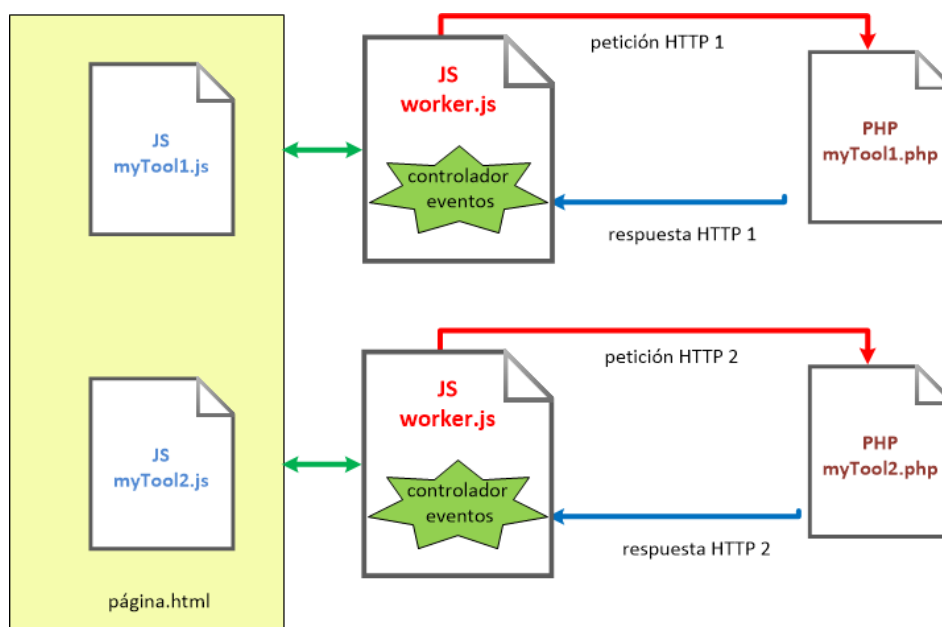


Ilustración 43 Esquema de funcionamiento de AJAX y programación multihilo

El tercer problema asociado al uso de AJAX apareció como consecuencia de emplear programación multihilo y de renovar periódicamente el identificador de sesión para mejorar la seguridad de la aplicación.

Las sesiones son un mecanismo ideado para solventar el problema de que HTTP sea un protocolo sin estado. Esto es; que una vez el servidor envía la respuesta al cliente, finaliza el proceso y elimina toda la información de la petición HTTP. No se conservan, pues, datos sobre el usuario entre visitas a las páginas del servidor.

Las sesiones son gestionadas por el servidor de aplicaciones¹³⁷ que almacena la información en archivos físicos o en una base de datos. Al iniciar una sesión, el servidor asigna automáticamente un ID (SID o *session ID*) al usuario que permite identificarlo de forma unívoca. Este identificador se puede transmitir al cliente de dos maneras: a través de la *QUERY_STRING*¹³⁸ o mediante una *cookie* (por defecto). A partir de este instante, cada vez que el usuario se conecta al servidor, se identifica enviando la *cookie* con el SID.

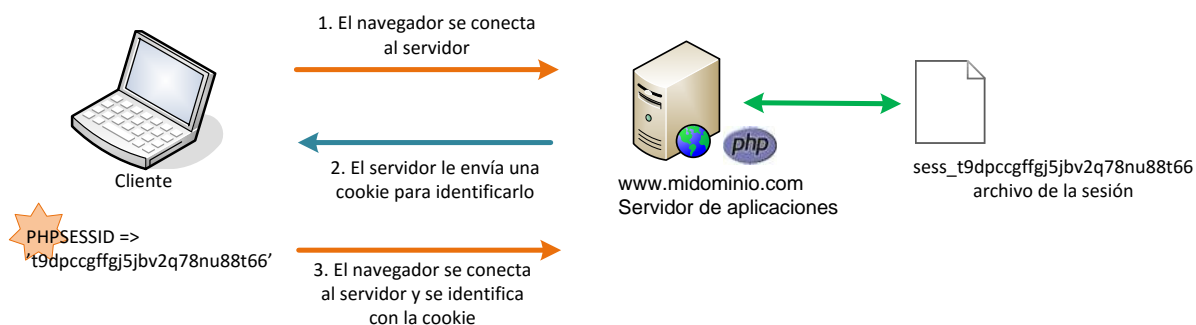


Ilustración 44 Mecanismo de identificación de una sesión de servidor

¹³⁷ PHP, JEE, Ruby...

¹³⁸ Porción de la dirección web (URL) que contiene datos que no pertenecen a la ruta del archivo. Se ponen al final de la URL tras un signo interrogativo y está formado por parejas clave-valor. Ejemplo:
http://www.ejemplo.cat/ruta?clave=valor

A partir de este esquema, se deduce que una de las mayores vulnerabilidades de este mecanismo es el robo del identificador de la sesión. Esta técnica consiste en suplantar la identidad del cliente copiando en otro ordenador la información del SID. Para evitarlo existen diferentes técnicas como el uso de comunicaciones seguras SSL, guardar en la sesión información extra para identificar al usuario, propagar las sesiones únicamente por *cookies* HTTP¹³⁹, codificar la información, almacenar la información en BD o en carpetas seguras y renovar periódicamente el SID. Este último punto fue especialmente problemático.

En un entorno de programación multihilo, cuando un hilo se conecta al servidor se identifica con el SID de la sesión. Dado que la renovación del SID se produce aleatoriamente cuando se establece una conexión, puede ocurrir que al producirse varias conexiones simultáneas cambie el SID de una de ellas. A partir de este momento, el resto de las conexiones concurrentes serán rechazadas por utilizar el antiguo SID que ha caducado. Es un problema complicado¹⁴⁰. Tras mucha investigación y pruebas se implementó una solución de compromiso: regenerar el SID solo cuando se carga una nueva página, se cierra, o se modifica el contenido de la barra.

Como se ha podido constatar, el uso de la tecnología AJAX no está exento de problemas cuando se requieren prestaciones más allá de las aplicaciones habituales.

Para facilitar la instalación en la barra de nuevas herramientas y su distribución, se optó por empaquetar todos los archivos en una única carpeta. Aunque el contenido puede diferir de una herramienta a otra, todas tienen una estructura típica con los mismos archivos y directorios. Una herramienta tipo consta de:

- **dictionary**, carpeta que contiene los diccionarios de idiomas con los mensajes que aparecen en la UI. Están en formato JSON y el nombre tiene la forma "dictionary.xx.json" donde "xx" es el código ISO 639-1 de dos letras¹⁴¹
- **image**, carpeta que contiene los iconos e imágenes, en formato SVG, usadas por la herramienta
- **app.json**, archivo de configuración de la aplicación
- **myTools.css**, archivo con la hoja de estilos que se aplica a la herramienta
- **myTools.js**, archivo con la implementación en javascript del código del cliente
- **myTools.php**, archivo con la implementación en PHP del código del servidor
- **worker.js**, archivo escrito en javascript con el código del hilo que gestiona la comunicación cliente-servidor

El nombre de la carpeta y de los archivos con la hoja de estilos, el archivo javascript con el código del cliente y el PHP con el del servidor coinciden. La presencia de estos archivos no es obligatoria. De hecho, en algunas herramientas no están presentes los archivos CSS (subrayado) o PHP (calculadora o portafolio) porque no eran necesarios.

¹³⁹ Cookies generadas por el servidor.

¹⁴⁰ <http://forums.devshed.com/php-development-5/regenerating-session-ids-safely-concurrent-request-921947.html>
<http://stackoverflow.com/questions/29770389/session-id-regeneration-not-working-for-concurrent-requests>
<http://stackoverflow.com/questions/15686572/two-simultaneous-ajax-requests-wont-run-in-parallel>
<http://stackoverflow.com/questions/5133004/php-sessions-and-ajax-request-from-injected-js-code?rq=1>

¹⁴¹ http://www.loc.gov/standards/iso639-2/php/code_list.php

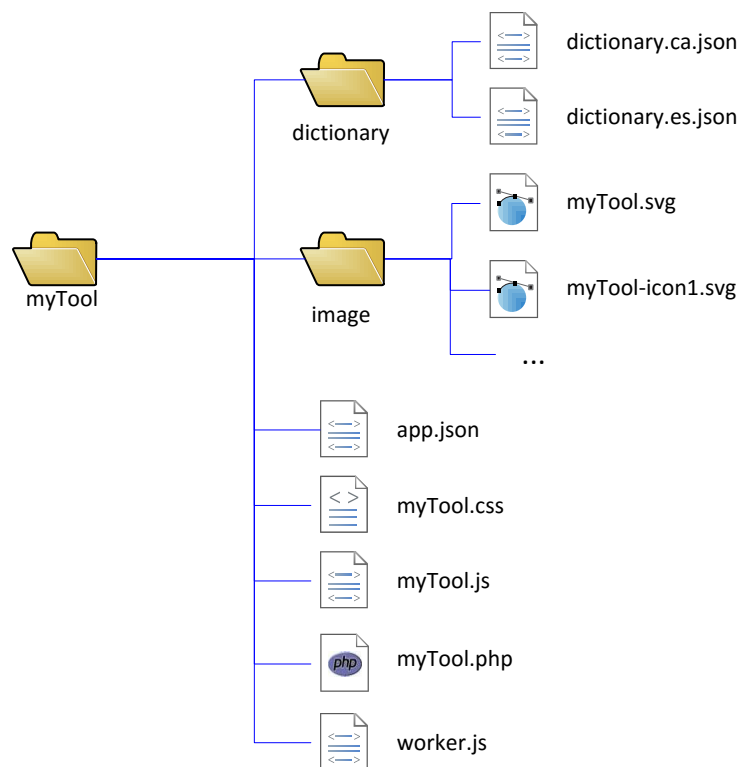


Ilustración 45 Estructura de archivos típica de una herramienta

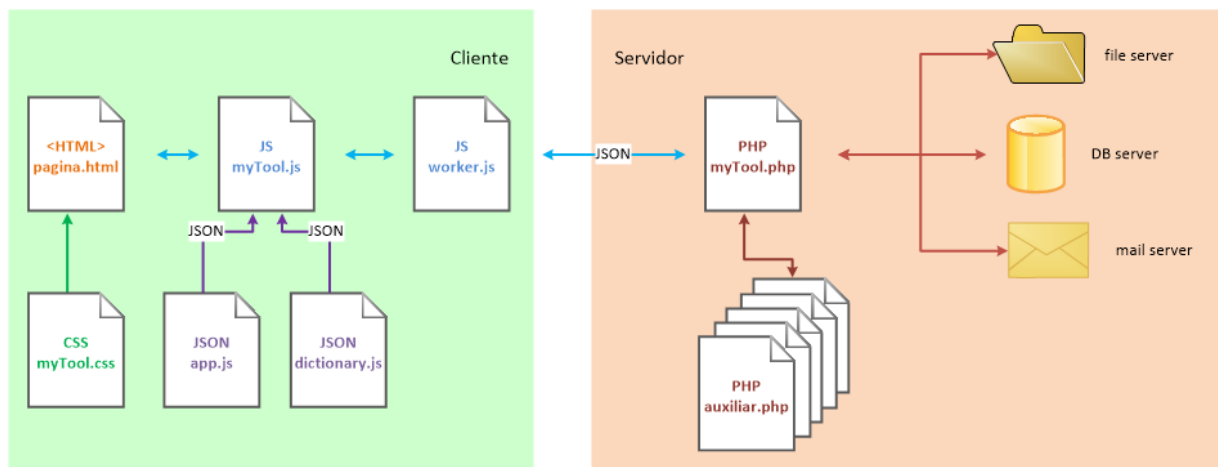


Ilustración 46 Relación de los archivos con la arquitectura cliente-servidor

Como ya se ha comentado anteriormente, existen tres archivos fundamentales para el funcionamiento de la herramienta.

- **worker.js**
- **myTool.js**
- **myTool.php**

En **worker.js** se implementa el hilo de ejecución secundario encargado de gestionar la comunicación entre el cliente y el servidor. Está compuesto de una variable, en la que se almacena la conexión, un método para enviar información, otro para recibirla, uno para tratar errores, así como los controladores de eventos correspondientes. El envío de mensajes al hilo principal se hace a través del método *postMessage()* de los *workers*.

```

var myWorker = (function () {
    var myConnection = new XMLHttpRequest();
    myConnection.addEventListener('error', errorOnAjaxCommunication);
    myConnection.addEventListener('load', receiveAjaxMessage);
    var ajaxServerFile;

    function sendAjaxMessage(e) {
        if (!myConnection) return false;
        var myjson = JSON.parse(e.data);
        ajaxServerFile = myjson.ajaxServerFile;
        var mytype = myjson.type;
        var myForm = null;
        if (myjson.form !== null) {
            myForm = new FormData();
            for (var myField in myjson.form) {
                myForm.append(myField, myjson.form[myField]);
            }
        }
        myConnection.open(myForm === null ? "GET" : "POST", ajaxServerFile);
        myConnection.responseType = mytype;
        myConnection.send(myForm);
    };

    function receiveAjaxMessage(e) {
        var data = e.target;
        if (data.status !== 200) {
            errorOnAjaxCommunication();
        }
        postMessage({
            "ajaxServerFile": ajaxServerFile,
            "response": data.response
        });
    };

    function errorOnAjaxCommunication(e) {
        console.log(e);
        postMessage(e);
    };

    return {
        sendAjaxMessage: sendAjaxMessage
    };
})();

addEventListener('message', myWorker.sendAjaxMessage);

```

Código 39 Archivo **worker.js**

El archivo **myTool.js** es el archivo donde se gestiona la interfaz de usuario (control y vista) de la herramienta. Los métodos implementados se especializan para realizar diferentes funciones:

- Gestión de la interfaz de usuario: *createAppButton*, *createAppSaveButton*, *createAppWindow*, *createContextItemMenu*, *showAppWindow*
- Gestión de la comunicación con el hilo (*worker*): *error*, *receiveWorkerMessage*, *sendWorkerMessage*.
- Inicialización y finalización: *closeApp*, *getAppParameter*, *initApp*, *isClosed*, *loadAppCSS*, *loadConfig*, *readLanguagePreference*
- Guardar/ recuperar información del servidor: *getPortfolioData*, *restoreFromDB*, *returnPortfolioData*, *saveToDB*

Los métodos se implementan mayoritariamente en todas las herramientas. Aunque en algunos casos, por las características del módulo, se prescinde de ellos como ocurre en la calculadora que no guarda ningún dato en el servidor. Además de los métodos anteriores, las herramientas implementan otras rutinas necesarias para el funcionamiento de la aplicación.


```

var myTool = (function (){
// verificar compatibilidad con AJAX
if(!XMLHttpRequest) throw new Exception("The browser isn't compatible with AJAX El navegador no
és compatible amb AJAX");
// variables comunes
var appWorker, appConfig, appDictionary, scrollNode, nodeArticle;
// variables privadas de funcionamiento interno
...

// funciones comunes del módulo
function closeApp(){ ... }
function createAppButton(){ ... }
function createAppSaveButton(){ ... }
function createAppWindow(){ ... }
function createContextItemMenu(myIcon, myTitle){ ... }
function error(e){ ... };
function getAppParameter(myparam){ ... }
function getPortfolioData(portfolio){ ... }
function initApp(mypath){ ... }
function isClosed(){ ... };
function loadAppCSS(){ ... }
function loadConfig(myObj){ ... }
function readLanguagePreference(){ ... }
function receiveWorkerMessage(e){ ... }
function restoreFromDB(json){ ... }
function returnPortfolioData(myData){ ... }
function saveToDB(){ ... };
function sendWorkerMessage(myfile){ ... }
function showAppWindow(){ ... }
...
// funciones del módulo
...

// Funciones públicas
return {
    initApp: initApp,
    closeApp: closeApp,
    isClosed: isClosed,
    getPortfolioData: getPortfolioData,
    getAppParameter: getAppParameter
};
})();

// carga del módulo
mySession.myTool = myTool;
mySession.myTool.initApp("/studentTool/myTool/",mySession);

```

Código 40 Arquitectura del módulo cliente de una herramienta implementado en javascript

Las dos últimas líneas de código de cada módulo son esenciales para el funcionamiento de las herramientas. La primera sirve para añadir el módulo a la barra de herramientas; técnicamente se añade una nueva propiedad (variable) a la barra en la que se guarda el módulo. Y la segunda pone en funcionamiento la herramienta.

El código del archivo **myTool.js** está implementado utilizando programación orientada a objetos y siguiendo un patrón modular en javascript. Este tipo de implementación emula el comportamiento de una clase¹⁴² y permite aislar el entorno de ejecución del módulo del resto de rutinas presentes en la página web. De esta forma se evitan interferencias entre las diferentes herramientas que trabajan simultáneamente.

El patrón modular se basa en dos características de javascript: la primera es que el ámbito de ejecución de una variable es la función donde está definida, y la segunda en que las funciones son también objetos. Si introducimos el código dentro de una función anónima¹⁴³ que se ejecute de forma inmediata¹⁴⁴ y lo asignamos a una variable, se obtiene un módulo. A continuación se muestra un patrón típico de un módulo en javascript¹⁴⁵:

¹⁴² En javascript no existe una sintaxis para definir clases al estilo de java o C++ hasta la versión 6 de *ecma-script* que se hizo oficial en verano de 2015, fecha posterior al desarrollo de este trabajo.

¹⁴³ Función sin nombre que se define y se utiliza de forma puntual. Su sintaxis es de la forma *function (argumentos) {...}*

```

var myModulo = (function (){
    var propiedadPrivada1, propiedadPrivada2,...;
    ...
    function funcionPrivada1(){ ... };
    function funcionPrivada2(){ ... };
    ...
    return{
        propiedadPublica1: propiedadPrivada1,
        funcionPublica1: funcionPrivada1
    }
})();

```

Código 41 Patrón de programación modular en javascript

La lógica de negocio de cada herramienta, el modelo en la arquitectura MVC, se implementa y gestiona desde el archivo **myTool.php** que se ejecuta en el servidor. Es un archivo implementado en PHP siguiendo el paradigma de la programación orientada a objetos. Dado que el trabajo principal de la aplicación se implementa en el navegador del usuario, el servidor se limita a gestionar la sesión del usuario, seguridad, validar datos e interactuar con la base de datos. Para simplificar el desarrollo, las tareas más comunes se han implementado en sus propias librerías desarrolladas *exprofeso* o se ha recurrido a terceros como en el caso del correo electrónico, que tiene licencia GNU¹⁴⁶.

La estructura de este archivo se inicia con la carga de las librerías de funciones, encargadas de realizar las tareas más comunes. A continuación se comprueba la validez de la sesión y en caso afirmativo se abre una conexión con la base de datos. En función de la orden recibida del cliente (*restore*, *sabe* y *delete* en el ejemplo), se comprueba la validez de los datos y se ejecuta la consulta correspondiente en la base de datos. El proceso puede acabar de tres maneras: que todo vaya bien en cuyo caso envía el valor 1 al cliente; -1 si falla y un mensaje en caso de excepción.

```

<?php
try{
    require_once('../llibreria/generalUserFunctions.php');
    require_once('../config.php');
    require_once('../session/session.php');
    require_once('../session/vista.php');
    require_once('../llibreria/database.php');
    require_once('../llibreria/warningHandler.php');
    require_once('../session/validar.php');
    require_once('../llibreria/xifrat.php');

    try{
        $err code = null;
        $err msg = null;
        if(!StudentToolSession::existsSession(false)){
            StudentToolSession::closeSession();
            throw new Exception("sessionError Highlight");
        }else{
            $myUserId = StudentToolSession::getUserId();
            $myCourse = StudentToolSession::getCourse();
            $myUrl = Validar::validarUrl("Url");
            $cn = new PDO(DB PDO CONNECTION,DB USER,DB PASS, array(PDO::ATTR ERRMODE =>
PDO::ERRMODE EXCEPTION));
            if(!empty($ GET['restore'])){
                $sth = $cn->prepare("call loadHighlight(:idUser,:url)");
                $sth->bindParam(':idUser', $myUserId, PDO::PARAM_INT);
                $sth->bindParam(':url', $myUrl,PDO::PARAM_STR);
                $result = $sth->execute();
                if($result){
                    $highlightDocument = $sth->fetchAll(PDO::FETCH_NUM);

```

¹⁴⁴ Función que el programa compila, ejecuta y una vez finalizada, se desecha de la memoria. Este comportamiento es contrario al habitual de las funciones que se conservan en memoria hasta la finalización del programa.

¹⁴⁵ Más información en:

<http://www.etnassoft.com/2011/04/11/el-patron-de-modulo-en-javascript-en-profundidad/>

<http://addyosmani.com/resources/essentialjsdesignpatterns/book/#modulepatternjavascript>

<http://www.adequatelygood.com/JavaScript-Module-Pattern-In-Depth.html>

¹⁴⁶ <http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>

```

        $sth->closeCursor();
    }
    echo count($highlightDocument)>0? $highlightDocument[0][0]:"";
} else if (!empty($_GET['save'])) {
    $innerHtml = Validar::validarHtml("html", "POST", $myUrl);
    $sth = $cn->prepare("call saveHighlight(:idUser, :course, :url, :html, @state)");
    $sth->bindParam(':url', $myUrl, PDO::PARAM_STR);
    $sth->bindParam(':course', $myCourse, PDO::PARAM_STR);
    $sth->bindParam(':idUser', $myUserId, PDO::PARAM_INT);
    $sth->bindParam(':html', $innerHtml, PDO::PARAM_STR);
    $result = $sth->execute();
    $sth = $cn->query("select @state;");
    $state = $sth->fetch()[0]; // siempre devuelve -1 o 1
    echo json_encode($state);
} else if (!empty($_GET['delete'])) {
    $sth = $cn->prepare("call deleteHighlight(:url, :idUser, @state)");
    $sth->bindParam(':url', $myUrl, PDO::PARAM_STR);
    $sth->bindParam(':idUser', $myUserId, PDO::PARAM_INT);
    $result = $sth->execute();
    $sth = $cn->query("select @state;");
    $state = $sth->fetch()[0]; // siempre devuelve -1 o 1
    echo json_encode($state);
}
$cn = null;
}
} catch (Exception $ex) {
    $err_code = $ex->getCode();
    echo $myVista->showRequestedHtml("user-management-access", false, true, true, true,
    $err_code);
}
} catch (Exception $ex) {
    echo "<pre>"; var_dump($ex); echo "</pre>";
}
}

```

Código 42 Ejemplo de archivo de la herramienta en el servidor

Uno de los procesos más críticos para la implementación del servidor fue la validación de los datos recibidos desde cualquier fuente de datos (formularios HTML, *cookies*, sesiones, *querystrings*, etc). El problema radica en que, si no se establecen las medidas necesarias, es posible el ataque de la web por parte de terceros. Para que esto no suceda, el programa incorpora una librería de validación que comprueba cualquier tipo de dato enviado al servidor; desde números, a archivos solicitados o sesiones. El proceso de validación de una variable comienza por la aplicación de filtros predefinidos en PHP¹⁴⁷ a través de las funciones *filter_var()* y *filter_input()*. Estos filtros permiten eliminar las etiquetas HTML y caracteres problemáticos para evitar ataques de inyección de HTML, SQL, etc al servidor. El siguiente paso consiste en comprobar que los tipos de datos se ajustan al formato esperado (si es un número, una URL, un *string*, etc) incluyendo limitaciones como rangos permitidos, listas de valores, etc. En caso necesario, se hacen comprobaciones extra como mirar si existe la página del curso antes de guardar las notas o resúmenes asociados.

A lo largo de este capítulo, se ha expuesto la arquitectura desarrollada para implementar nuevas herramientas y algunas de las dificultades encontradas. A continuación se expondrá el proceso necesario para añadir una nueva herramienta a la barra. El proceso consta de tres pasos:

- En primer lugar, añadir la carpeta con la herramienta a la carpeta con la barra
- A continuación, modificar el valor de la constante `STUDENTTOOL_MODULES` del archivo **config.php** y añadir el nombre de la aplicación (coincide con el de la carpeta y archivos principales)
- Finalmente, hay que asegurarse de que en el archivo **myTool.js** se han incluido las líneas que añaden la herramienta a la barra y la inicializan

¹⁴⁷ <http://php.net/manual/es/ref.filter.php>

8.2 Base de datos

Dado que la mayoría de las herramientas de la barra necesitan almacenar información en el servidor para su uso posterior, se utiliza una base de datos multiusuario con capacidad para manejar grandes cantidades de información.

El diseño de la base de datos se simplificó al máximo, reduciendo el número de tablas al mínimo imprescindible. De esta manera se simplifican las consultas y se optimiza la velocidad de acceso. La estructura de la base de datos sigue el diseño de un diagrama de estrella, con una tabla central, la de los usuarios, que está relacionada con todas las demás. El resto de las tablas, se corresponden cada una a una herramienta de la barra. En ellas, sólo se guardan los datos imprescindibles y se ha renunciado a emplear un registro histórico de la actividad de los usuarios.

Esto generó muchas dudas durante la fase de diseño; pues se cuestionó si la trazabilidad de los datos de los alumnos sería buena manera para evaluar su actividad. Como el programa no incluye una aplicación en la que el formador pueda realizar un seguimiento del usuario, no queda justificado incluir los campos en las tablas. Así se simplifica la implementación de la base de datos y se aumenta la velocidad de respuesta.

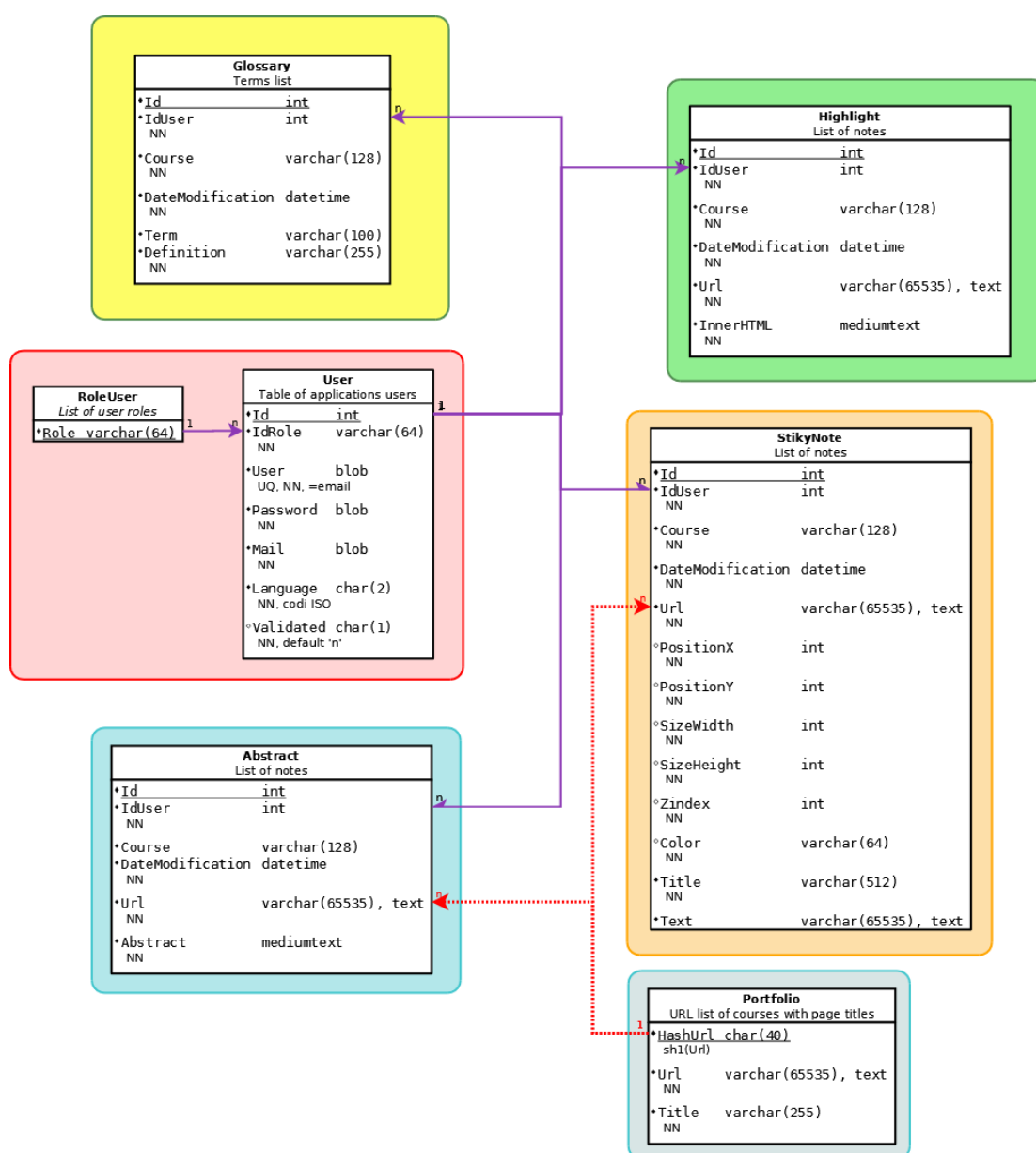


Ilustración 47 Diagrama entidad relación de la BD de la aplicación

El acceso a los datos de las tablas se hace a través de procedimientos (*procedures*) y funciones (*functions*) programados dentro de la base de datos. Aunque es más complejo de implementar que acceder directamente con consultas SELECT, INSERT, UPDATE o DELETE, aporta dos ventajas fundamentales:

- La primera es que al guardarse los procedimientos dentro de la base de datos, ya están compilados y se ejecutan más deprisa porque el SGBD¹⁴⁸ no pierde tiempo analizando la sintaxis de la consulta, validándola y compilándola antes de ejecutarla.
- La segunda, y más importante es la seguridad. Cuando se permite la ejecución de consultas directamente, normalmente el usuario dispone de permisos de acceso a toda la base de datos. El peligro radica en que un usuario malicioso, una vez conseguido el usuario y contraseña de acceso, podría ejecutar cualquier consulta en la base de datos y conseguir cualquier información. Sin embargo, si se utilizan procedimientos como se ha utilizado en este proyecto, lo único que puede ejecutar son estas funciones y no se puede acceder libremente a los datos de las tablas.

```
delimiter $$

create procedure loadAbstract(
  _IdUser int,
  _Url text
)
begin
  -- retrieves the list of abstracts
  select ab.`Abstract`
  from `StudentTool`.`Abstract` ab
  where ab.`IdUser`= IdUser and ab.`Url`= Url;
end $$

create procedure saveAbstract(
  IdUser int,
  Course varchar(128),
  _Url text,
  _Abstract mediumtext,
  PageTitle varchar(255),
  OUT result int
  -- result is -1 if process fails and 1 if it works
)
begin
  declare _iid int;
  declare HashUrl char(40);
  set result=-1;
  -- verify if user exists
  select Id into iid from User where Id= IdUser limit 1;
  if iid is not null then
    set _iid=null;
    -- verify if abstract text exists, ID is null
    select Id into iid from `StudentTool`.`Abstract` where `IdUser`= IdUser and `Url`= Url;
    if _iid is null then
      -- insert new
      INSERT INTO `StudentTool`.`Abstract`
      (`IdUser`,`Course`,`DateModification`,`Url`,`Abstract`)
      VALUES
      ( IdUser, Course,now(), Url, Abstract);
      -- tractament de les url
      select `HashUrl` into _HashUrl from `StudentTool`.`Portfolio` where `HashUrl` = sha1(_Url);
      if _HashUrl is null then
        insert into `StudentTool`.`Portfolio` (`Url`,`Title`) values ( Url, PageTitle);
        end if;
    else
      -- update
      UPDATE `StudentTool`.`Abstract`
      SET `DateModification`= now(), `Abstract`=_Abstract
      WHERE `IdUser`= IdUser and `Url`= Url;
    end if;
  if row_count()>0 then
    set _result=1;
  
```

¹⁴⁸ Sistema gestor de la base de datos, programa que gestiona el acceso a la información; en este caso MySQL.

```
end if;  
end if;  
end $$  
delimiter ;
```

Código 43 Ejemplos de *PROCEDURE* en MySQL

Como medida extra de seguridad en la base de datos, y para cumplir con la legislación de protección de datos se encriptaron las columnas de la tabla de usuarios que contienen información sensible relacionada con la privacidad de los usuarios: la contraseña y el correo electrónico.

8.3 Barra de herramientas y gestión de usuarios

La barra de herramientas proporciona el soporte necesario para empaquetar las herramientas y darles un aspecto uniforme. Bajo el aspecto de una caja de herramientas, se posicionan los iconos de las diferentes aplicaciones que el usuario puede utilizar para personalizar el curso.



Ilustración 48 Barra de herramientas

Dado que al curso pueden acceder simultáneamente un número indeterminado de usuarios, fue necesario implementar un sistema de gestión que garantizara la privacidad e independencia de los contenidos de cada individuo. Ante esta tesitura, y dado que uno de los requisitos de desarrollo de la aplicación era minimizar los costes incluidos los de mantenimiento, se optó por crear un sistema de gestión de usuarios autónomo. De esta manera, el administrador no es una pieza fundamental para su funcionamiento y puede despreocuparse del día a día de los usuarios. Esto complicó el desarrollo, pues había que contemplar todas las incidencias posibles y programarlas para que el usuario pudiera solucionarlas por sí mismo. En resumen, el motor de gestión de usuarios tenía que hacer lo siguiente:

- Dar de alta usuarios
- Dar de baja usuarios
- Cambiar la contraseña
- Editar los datos del perfil
- Recordar datos de acceso

No hay que olvidar que el objetivo principal de la barra es el de dar soporte a las aplicaciones. Para ello hubo que diseñar un sistema que integrara las herramientas de forma sencilla.

La interfaz de usuario se divide en dos partes: la gestión de usuarios y la barra propiamente dicha.

La gestión de usuarios es similar a la de cualquier otra red social.

Ilustración 49 Pantalla de inicio de sesión y acceso a las herramientas

La pantalla de inicio de sesión es la puerta de acceso a la barra de herramientas. Es necesario validarse como usuario para acceder a la barra y disfrutar de la aplicación. Ofrece un nutrido grupo de opciones de autogestión e información acerca de la aplicación y la política de *cookies*. En la primera se ofrece información legal, técnica y de autoría sobre las herramientas. Mientras, en la segunda, se ofrece información sobre la política de privacidad tal como obliga la legislación vigente¹⁴⁹.

Otros enlaces remiten a los mecanismos necesarios para recuperar la contraseña, el nombre de usuario o para realizar el trámite final en el alta o modificación de una cuenta de usuario. En todos los casos, se accede a una pantalla en la que el usuario introduce su cuenta de correo, que utilizó para darse de alta y en la que recibirá un correo con información para realizar las gestiones solicitadas.

Ilustración 50 Pantalla para introducir el correo

Un último enlace ofrece la oportunidad de darse de alta como nuevo usuario. Desde él, se accede a un formulario en el que se han de rellenar algunos datos: nombre de usuario, idioma, contraseña, correo electrónico y el botón de confirmación de las condiciones de uso. Como particularidad, no es posible marcar ni enviar los datos hasta que se han "leído" todas las condiciones de uso.

Ilustración 51 Pantalla para introducir un nuevo usuario

¹⁴⁹ [Real Decreto-ley 13/2012](#), de 30 de marzo y [Directiva 2009/136/CE](#), del Parlamento Europeo y del Consejo, de 25 de noviembre de 2009

La mecánica y funcionamiento de todas estas pantallas queda reflejado en el siguiente diagrama:

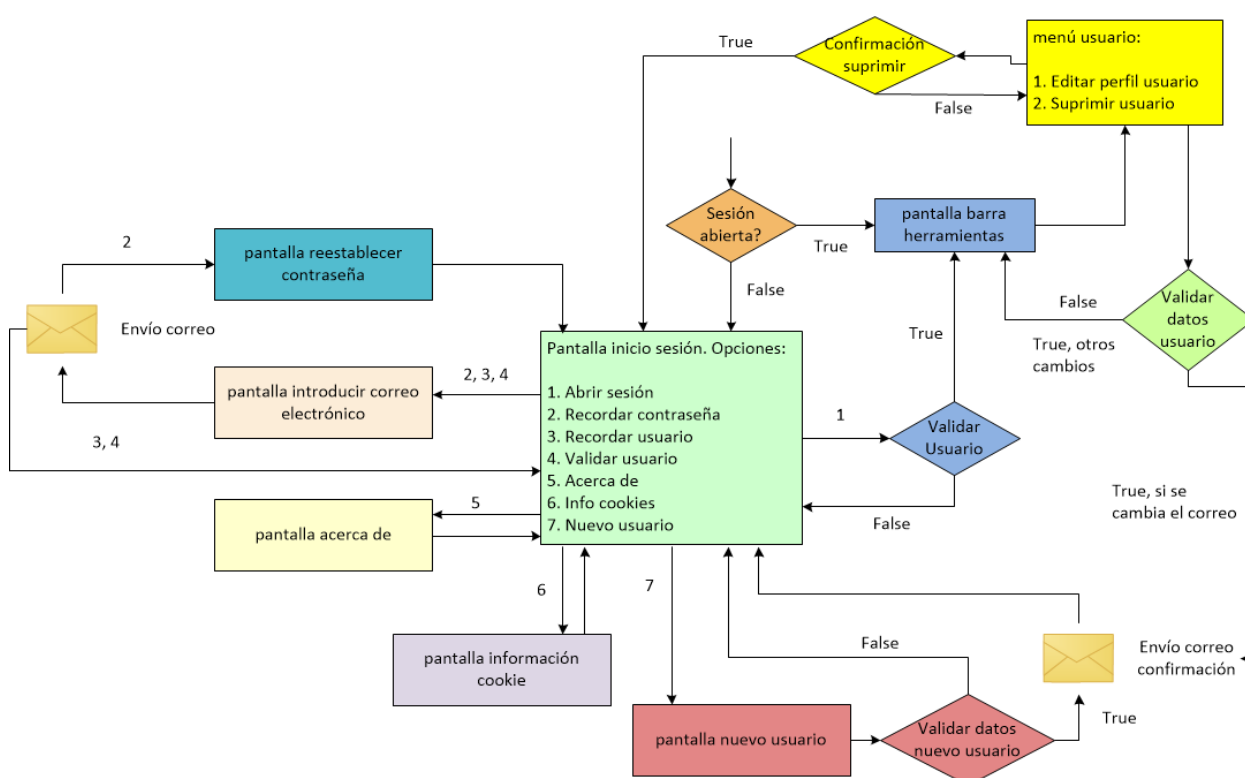


Ilustración 52 Esquema de la interfaz de usuario para gestionar las sesiones

Una vez se ha confirmado que el usuario es válido, se carga la barra de herramientas con los iconos de las aplicaciones. El último icono (ilustración 50) sirve para acceder al menú del usuario. Al presionarlo aparece un menú contextual con dos opciones: editar el perfil y eliminar la cuenta de usuario.



Ilustración 53 Menú contextual de usuario

Editar perfil

Usuario

Dirección de correo

Idioma es ▾

Deja la contraseña en blanco si no la quieres cambiar.

Contraseña

Repítela

Aplicar los cambios

Si has cambiado la dirección de correo, recibirás un correo para confirmarlo a la nueva dirección.

Ilustración 54 Pantalla para editar el perfil

Es posible modificar la contraseña, el idioma y la cuenta de correo asociada al perfil. El nombre de usuario no se puede cambiar. Se evita así que se pueda introducir un nombre que ya exista en la BD¹⁵⁰.

Los datos de los usuarios se almacenan en una tabla de la base de datos del programa, al igual que el resto de información del resto de aplicaciones. La contraseña y el correo electrónico se guardan encriptados por motivos de seguridad.

La implementación de la barra sigue el esquema cliente-servidor del resto de herramientas e implementa los mismos métodos que se explicaron al hablar de la arquitectura de la aplicación.

Una de las mayores diferencias con el resto de herramientas está en la manera en que se generan las pantallas. En otros casos, se generan íntegramente con javascript y los textos se extraen de un archivo de configuración. Pero aquí, esto no resultaba práctico. Había demasiado texto y las pantallas contenían muchas etiquetas y formularios. Y además, si se hubiera querido alterar la estructura y diseño de las pantallas, habría supuesto un trabajo complicado. Por ello, se optó por incluir todo el texto de las pantallas dentro de un archivo HTML a modo de plantilla. En concreto, uno por cada idioma en que esté la herramienta.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>User Management</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
</head>
<body>
<div id="button-return" role="button" tabindex="0" title="Atrás">&lt;</div>
<div id="button-close" role="button" tabindex="0" title="Cerrar">X</div>
<div id="button-minimize" role="button" tabindex="0" title="Minimiza">&#8744;</div>
<div id="button-maximize" role="button" tabindex="0" title="Maximiza">&#8743;</div>

<div id="user-management-access">
<p class="no-see">Cuenta confirmada. Gracias por darse de alta en nuestro servicio.</p>
<form action="" id="open-session-form">
<label>Usuario <input type="text" name="user" size="15" maxlength="64" autocomplete="on"
placeholder="Usuario" /></label>
<label>Contraseña <input type="password" name="password" size="15" maxlength="30"
placeholder="Contraseña" /></label>
<input type="submit" value="Inicia la sesión" />
<div class="access-forgot-new">
Has olvidado
<a href="account/password-reset" title="Inicio del proceso de recuperar contraseña">¿la
contraseña?</a>
<a href="account/remember-user" title="Recuperar el nombre de usuario">¿el nombre de
usuario?</a> .
<a href="account/validate-user" title="Validar el usuario">validar el usuario</a><br />
<a href="account/new-user" title="Iniciar el procedimiento para crear una nueva cuenta">Nueva
cuenta</a> .
<a href="app/info" title="Información sobre la aplicación">La aplicación</a>
</div>
</form>
<div id="cookies-policy">
Esta web utiliza cookies para mejorar la experiencia de navegación. Si sigues navegando
entendemos que la aceptas.
<a href="cookies/information" title="Política de cookies">Más información</a>
</div>
</div>

<div id="cookies-info">
<h2>Política de cookies</h2>
...
</div>

<div id="app-info">
<h2>Información</h2>
...
</div>
```

¹⁵⁰ Base de datos

```

<div id="account-reset-begin">
...
</div>

...

<div id="error-list">
<ul>
<li id="er-101">No se encuentra la carpeta con los diccionarios.</li>
...
</ul>
</div>

</body>
</html>

```

Código 44 Fragmento de la plantilla con el código de las pantallas de la barra

De esta manera, la edición y la posibilidad de añadir nuevos idiomas al programa se realizan de manera sencilla y funcional. El funcionamiento de la generación de la interfaz es conceptualmente más simple. El cliente solicita al servidor la pantalla correspondiente, y este, que previamente ha cargado en memoria el DOM con la plantilla, extrae el subárbol adecuado del `<div>` y se lo envía como HTML al navegador. Este último, sustituye el contenido del nodo del DOM de la barra y registrar los controladores de eventos para la nueva pantalla.

Este proceder no está exento de problemas. Uno de ellos, es que no es tan flexible como parece pues como hay que definir eventos para los elementos de la pantalla (botones, enlaces...), en la práctica hay que respetar la estructura. Si se elimina el nodo equivocado, la aplicación deja de funcionar. Respecto a los enlaces, hay que resaltar que éstos son meramente formales, se usan como un medio para que sea posible cumplir con los requisitos de accesibilidad y evitar tener que aplicar ARIA. En realidad, es javascript quien determina lo que hacen los enlaces.

Otra particularidad de la implementación es la adaptación de la altura de la barra de herramientas a su contenido (siempre que quepa en la ventana del navegador). Este es un problema que se presenta siempre que se quieren mostrar contenidos mediante capas que simulan ventanas. Dar unas dimensiones fijas funciona en la mayoría de casos, pero no en este, en el que hay contenidos tan dispares; desde barras de iconos hasta textos con varios párrafos. Se hizo necesario implementar una rutina que fuera capaz de calcular la altura necesaria de la ventana dado un ancho fijo. Una vez calculada, la altura se asignaba mediante una clase o aplicando el atributo *style* a la etiqueta con la pantalla. La gracia reside en que cuando se vuelve a la pantalla inicial se elimina el añadido anterior y vuelve a las dimensiones originales. Todo se basa en comprender el algoritmo que determina cómo se aplican las propiedades CSS a una etiqueta.

También se añadieron rutinas para minimizar la ventana y desplazarla por el navegador. Ante los conflictos de eventos¹⁵¹ que se producían al pulsar en zonas con texto, se optó porque sólo se pudiera desplazar la ventana pulsando sobre el margen exterior que rodea el texto.

La generación de la barra de herramientas con los iconos de las aplicaciones fue un proceso complejo que requirió de bastantes pruebas e investigación, pues suponía subir un escalón en la programación modular de la aplicación. Era preciso que cargara los archivos de javascript selectivamente y después incorporara las herramientas como *plugins* de la barra. Esto se consigue haciendo que al cargar la barra, el servidor envíe el código HTML con el contenido de esta y la información de los módulos que deben cargarse camuflada en una etiqueta `<div>`.

¹⁵¹ Ver una explicación más detallada en el apartado dedicado a las [notas](#).

```

<div id="student-tools-account-menu" lang="es">
<ul id="user-menu"></ul>
<div id="modules-studentTool"
data-modules=' [
{"file":"\\studentTool\\calculator\\calculator.js", "name": "calculator"},
{"file":"\\studentTool\\highlight\\highlight.js", "name": "highlight"},
{"file":"\\studentTool\\abstract\\abstract.js", "name": "abstract"},
{"file":"\\studentTool\\glossary\\glossary.js", "name": "glossary"},
{"file":"\\studentTool\\stikyNote\\stikyNote.js", "name": "stikyNote"},
{"file":"\\studentTool\\portfolio\\portfolio.js", "name": "portfolio"}
] '></div>
<div id="button-end-session" role="button" tabindex="0" title="Cerrar">X</div>
<div id="button-maximize" role="button" tabindex="0" title="Maximiza">^</div>
<div id="button-minimize" role="button" tabindex="0" title="Minimiza">v</div>
</div>

```

Código 45 HTML con el contenido de la barra de herramientas. Obsérvese la etiqueta <div> con la información de los módulos

A partir de la información en formato JSON contenida en el atributo "data-modules", el programa inyecta etiquetas HTML en el <head> de la página.

```

<head>
...
<link rel="stylesheet" type="text/css" href="/llum/css/reset.css" media="all">
<link rel="stylesheet" type="text/css" href="/llum/css/estilBasic.css" media="screen">
...
<script type="text/javascript" src="/llum/js/codi.js" defer=""></script>
<script type="text/javascript" src="/studentTool/session/session.js"></script>
<link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Open+Sans">
<link rel="stylesheet" type="text/css" href="/studentTool/session/session.css" media="screen">

<script src="/studentTool/calculator/calculator.js" type="text/javascript"></script>
<script src="/studentTool/highlight/highlight.js" type="text/javascript"></script>
<script src="/studentTool/abstract/abstract.js" type="text/javascript"></script>
<script src="/studentTool/glossary/glossary.js" type="text/javascript"></script>
<script src="/studentTool/stikyNote/stikyNote.js" type="text/javascript"></script>
<script src="/studentTool/portfolio/portfolio.js" type="text/javascript"></script>
<link rel="stylesheet" type="text/css" href="/studentTool/calculator/calculator.css">
<link rel="stylesheet" type="text/css" href="/studentTool/stikyNote/stikyNote.css">
<link rel="stylesheet" type="text/css" href="/studentTool/portfolio/portfolio.css">
<link rel="stylesheet" type="text/css" href="/studentTool/abstract/abstract.css">
<link rel="stylesheet" type="text/css" href="/studentTool/glossary/glossary.css">
</head>

```

Código 46 Extracto del código HTML de la etiqueta <head> modificado por el script

De esta manera, la edición y la posibilidad de añadir nuevos idiomas al programa se realiza de manera sencilla y funcional. Al no ser un proceso lineal, el navegador los va descargando sin un orden prefijado y cargando en memoria a medida que se bajan del servidor. A continuación se ejecutan las líneas de código que añaden la herramienta a la barra. Los propios *scripts* se encargan de añadir al DOM y bajar las hojas de estilo CSS necesarias para las herramientas y de cargar los iconos de la barra. Este es el motivo por el que no salen nunca las herramientas en el mismo orden en la barra. Es posible mantener el orden recolocando los elementos al final del proceso de carga implementando una rutina al efecto si se considerara necesario.

Conviene resaltar la peculiaridad de dos de los botones que aparecen en la barra de herramientas.

El primero es el icono con el menú del usuario. Este icono lo genera la barra de herramientas y siempre aparece ubicado al final de la barra.

El segundo es el icono de guardar los datos en la base de datos. Cuando se pulsa este botón, se inicia un proceso que lanza la orden de guardar los datos a todas las herramientas. El problema es que no todas tienen esta función (la calculadora por ejemplo). Imaginemos un caso extremo en que ninguna herramienta guarda datos en el servidor; ¡la presencia del botón de guardar resultaría ridícula! Ante el dilema de si hacer que la barra lo creara o fueran las

herramientas se optó por la segunda. Por ello, cuando una herramienta se añade a la barra, también incorpora el botón de guardar si es necesario. Para evitar que se repita muchas veces este proceso solo se realiza la primera vez.

Finalmente, una rutina se encarga de adaptar la anchura de la ventana a la cantidad de herramientas disponibles. Lógicamente, una variable controla el proceso para que sólo se carguen una vez los módulos en la página.

Cuando se cierra la barra de herramientas, bien porque se ha cerrado la sesión o se accede a las pantallas del menú del usuario, se dispara una rutina que se encarga de cerrar cada una de las herramientas individualmente. Esto es posible porque las herramientas se tratan como objetos-módulos que se guardan dentro un *array*, que es una variable de la barra. Así, se llama a cada una de las aplicaciones y se invoca el método *closeApp()* que guarda los datos y destruye los nodos con las ventanas, subrayados... que se hayan creado para que la apariencia del curso sea como si no se hubiera utilizado la herramienta.

La programación en el servidor, también es más compleja que en el resto de las herramientas. Esto se debe a que la generación de la vista está repartida entre el cliente y el servidor. De la implementación en el cliente, ya se ha explicado cómo funciona la creación de las pantallas a partir de los datos recibidos del servidor.

En el servidor están programadas las funciones que extraen el HTML de las pantallas de la plantilla. La vista funciona siguiendo los mismos principios que el programa de la plataforma de publicación de los cursos; esto es cargar un HTML en un árbol DOM y manipularlo según las necesidades.

La complejidad del controlador radica en que es el encargado del proceso de validación de las sesiones de usuario y de las gestiones de los perfiles. Para aumentar la seguridad de las sesiones, además de cambiar periódicamente el identificador de sesión y utilizar SSL, se utiliza una carpeta propia para guardar los archivos de la sesión con el SID, y el texto del archivo con los datos de la sesión se guarda encriptado¹⁵². De esta manera se dificulta el robo de la sesión. Además, entre los datos de la sesión se guarda información sobre el usuario y la fecha de caducidad de la sesión. El proceso para validar una sesión es el siguiente:

1. Que exista una cierta variable en la sesión
2. Que se puede descifrar su contenido
3. Que el objeto obtenido tiene las propiedades correctas
4. Que no haya caducado
5. Que los datos extras para identificar al usuario sean correctos

La lógica de negocios, el modelo, se centra en realizar la gestión de los usuarios. Esto es comunicarse con la base de datos para hacer las validaciones, altas, bajas y modificaciones de los datos y enviar los correos electrónicos a los usuarios para validar cuentas o recordar datos. La gestión del correo se realiza con una herramienta de código libre llamada PHPMailer¹⁵³.

8.4 Herramienta calculadora

La calculadora es una herramienta de apoyo a los cursos creada para facilitar la realización de ejercicios que incluyan cálculos numéricos. Nace con la vocación de ser útil, práctica, fácil de usar, exportable y ampliable en el futuro. Para ello, se ha puesto especial énfasis en tres aspectos: la interfaz de usuario, el manual de instrucciones y una programación modular y expandible que simplifique futuras ampliaciones.

¹⁵² Existe otra técnica para robar sesiones en servidores PHP que se aprovecha del hecho de que todos los archivos de sesión de las diferentes web alojadas utilizan por defecto la misma carpeta para guardarse. El sistema consiste en tener una web en el mismo servidor y leer el archivo de sesión de la web que se quiere *hackear*. Por defecto, los archivos de sesión se guardan en texto plano.

¹⁵³ <https://github.com/PHPMailer/PHPMailer>

Su funcionamiento se inspira en el de las calculadoras Hewlett Packard de la serie HP48¹⁵⁴ y, al igual que estas, utiliza notación polaca inversa (RPN)¹⁵⁵. En este sistema, se introducen en primer lugar los factores que intervienen en la operación y a continuación el operador.

Notación normal	Notación polaca inversa RPN
2 + 3	2 3 +
2 * (3 + 1)	2 3 1 + *
	3 1 + 2 *

Tabla 4 Comparación entre sistemas de notación de operaciones matemáticas

Notación normal	Notación polaca inversa RPN
2 + 3	2 3 + 5 -> resultado final
2 * (3 + 1)	2 3 1 + * 2 4 * 8 -> resultado final
	3 1 + 2 * 4 2 * 8 -> resultado final

Tabla 5 Ejemplos de ejecución de operaciones matemáticas en notación RPN

Los valores se van introduciendo en una pila de datos de la que se extraen para realizar las operaciones. En una operación de una variable, se toma un argumento de la pila y se devuelve el resultado. Análogamente, en una operación de dos variables se toman dos y se devuelve un resultado. Al principio es algo complicado y confuso, pero una vez los usuarios se han familiarizado con la mecánica es más cómodo, rápido y permite encadenar operaciones complejas con mucha facilidad.

8.4.1 Objetivo

Las ideas principales que se han tenido en cuenta para diseñar y programar la calculadora son las de ofrecer un complemento que ayude al usuario a seguir los cursos y explorar las posibilidades de javascript y SVG en el diseño de interfaces web.

En primer lugar, se pretende que el diseño gráfico sea agradable, sencillo y funcional. Que tenga el aspecto de una calculadora de mano, con combinaciones de colores que no resulten molestas a la vista, que todas las funciones sean fácilmente visibles y que se distingan bien los diferentes elementos que forman la calculadora.

En segundo lugar, la interfaz debe resultar intuitiva. Las teclas se agrupan según la naturaleza de las operaciones que realizan (teclado numérico, funciones matemáticas básicas, presentación en la pantalla, funciones trigonométricas, etc.). En la pantalla, todas las expresiones numéricas y mensajes se han de estar completos con todos sus dígitos y caracteres.

En tercer lugar se añadirá una ayuda interactiva, con explicaciones breves y sencillas, accesibles a todo tipo de públicos que sea visible cuando el usuario la necesite.

En cuarto lugar ha de incorporar las funciones básicas de una calculadora científica: opera con radianes y grados, realiza las operaciones básicas (suma, resta, multiplicación y división), las trigonométricas y sus inversas, trabaja con raíces, potencias y logaritmos. Además, es posible presentar los resultados en los modos habituales: NORM, FIX y SCI. Como extra, se añadió la facultad de trabajar con números complejos.

¹⁵⁴ https://en.wikipedia.org/wiki/HP_48_series

¹⁵⁵ https://es.wikipedia.org/wiki/Notaci%C3%B3n_polaca_inversa

El código empleado en la calculadora debe ser robusto. El programa debe filtrar los errores de introducción de datos, los problemas con el dominio de las funciones y el desbordamiento de las operaciones mostrando los correspondientes mensajes de alerta.

Finalmente, el código debe ser modular y flexible para poderlo someter a cambios puntuales sin necesidad de alterarlo completamente e incorporar nuevas operaciones fácilmente.

8.4.2 La interfaz de usuario

La interfaz del usuario está formada por la pantalla y el teclado. Son elementos que hay que diseñar con mucho cuidado y mimo porque de él depende que el usuario se familiarice rápidamente con su uso.

Para crear la interfaz se ha escogido SVG, un lenguaje de etiquetas que genera gráficos vectoriales, que presenta la ventaja de que admite interacción con javascript. De esta manera, modificar la interfaz es más fácil.

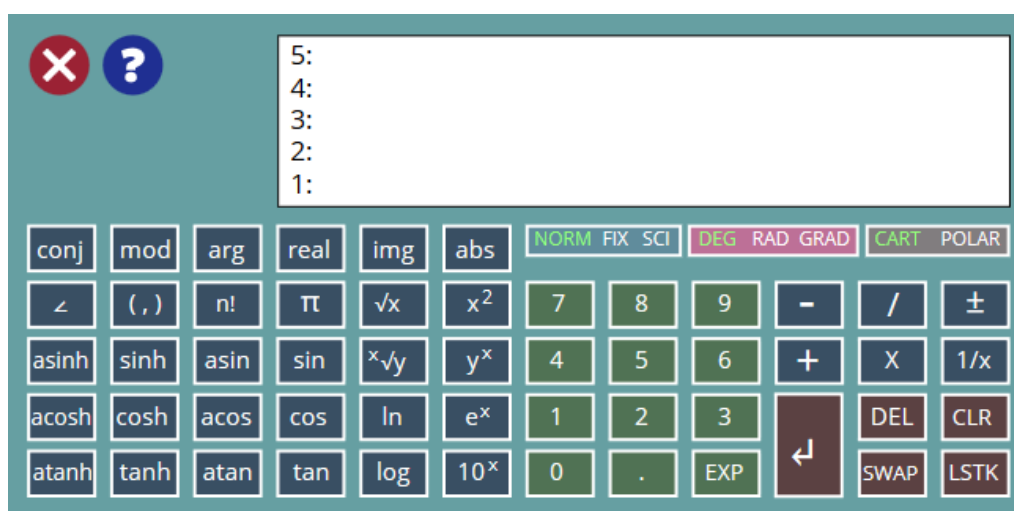


Ilustración 55 Interfaz de usuario de la calculadora

La pantalla dispone de tres modos de funcionamiento:

- En el modo normal, la pantalla tiene hasta cinco líneas para mostrar datos. En este modo se muestran cinco niveles de la pila.

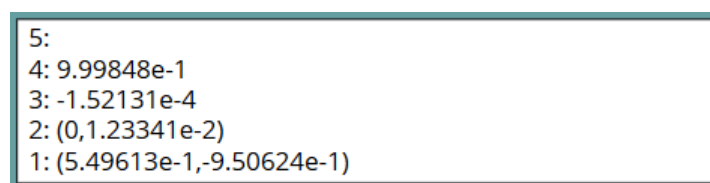


Ilustración 56 Modo normal de la pantalla

- En el modo de línea de comandos, tiene cuatro líneas para mostrar datos y la última sirve para introducir números. En este caso se muestran cuatro niveles de la pila.

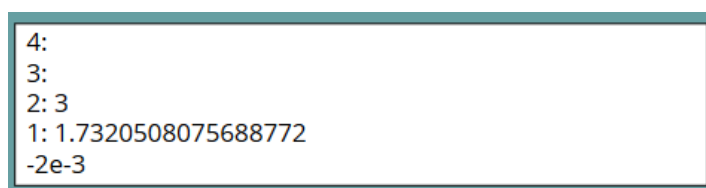


Ilustración 57 Modo de pantalla con línea de comandos

- En el modo de mensajes de error, también tiene cuatro líneas y la superior se utiliza para mostrar los mensajes de error. En este caso también se muestran cuatro niveles de la pila.

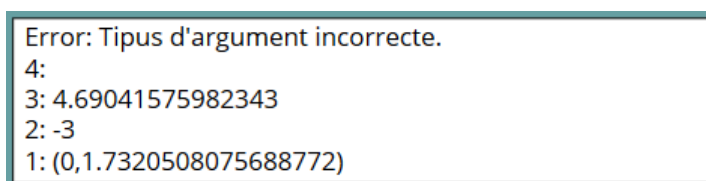


Ilustración 58 Modo de pantalla con mensaje de error

El número de líneas de la pantalla se ha establecido en cinco aunque se podrían añadir más modificando el parámetro correspondiente y la pantalla en el archivo SVG.

La cantidad máxima de caracteres que se aplica a la representación de los números reales y a las partes real e imaginaria de un número complejo está también definida por un parámetro. Por defecto, son 15 dígitos que coinciden con la precisión de javascript. En caso de utilizar modos de presentación en pantalla como FIX (número fijo de decimales) o SCI (notación científica) con un menor número de dígitos, una función se encarga de redondear los números y formatearlos.

La pantalla y la pila de números se gestionan de forma separada. Cada vez que se hace un cálculo se regeneran. De esta manera no se pierde precisión en los cálculos por culpa de los redondeos necesarios para hacer la presentación.

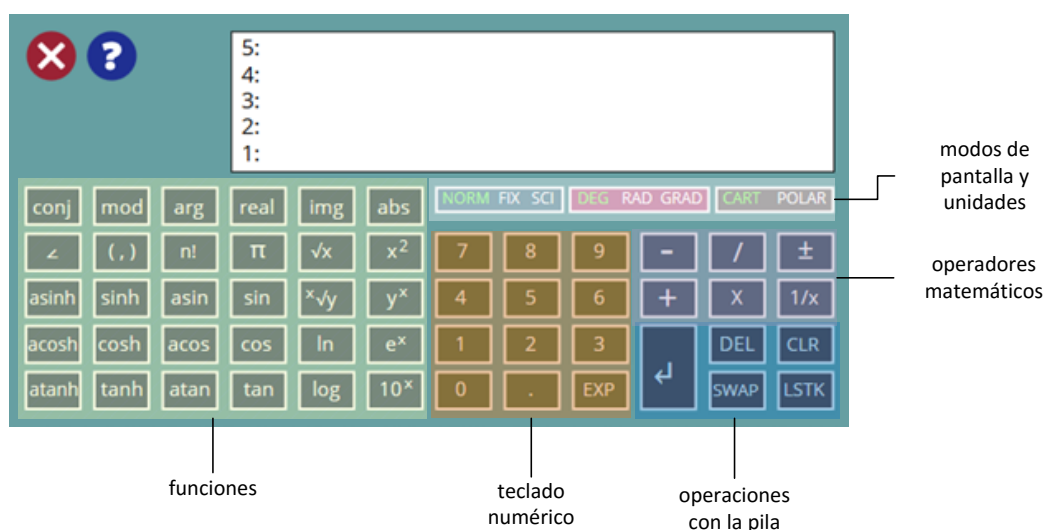


Ilustración 59 Áreas funcionales del teclado

El teclado está dividido en cinco áreas funcionales pensadas para facilitar el uso y localización de las funciones de la calculadora. Las teclas se han repartido con la idea de que se ajustaran a la distribución estándar. Con ello se pretende

que el teclado resulte más familiar a los usuarios. Para evitar ambigüedades, cada tecla realiza una única función. El problema es que dificulta la distribución lógica de las teclas; por ejemplo, en la segunda fila de las funciones se mezclan teclas para introducir números complejos, con el número pi, y funciones cuadráticas. Es una solución de compromiso que intenta ser lo más clara posible para el usuario.

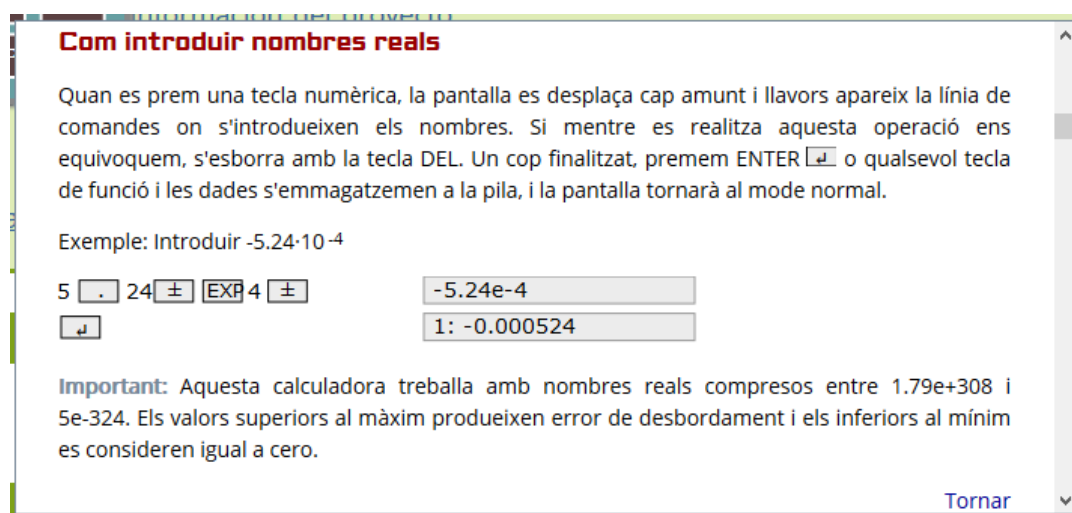
En la parte central está el teclado numérico estándar con las cifras decimales, la coma decimal¹⁵⁶ y el exponente. A su derecha se encuentran los operadores matemáticos comunes (suma, resta, multiplicación y división), el operador de cambio de signo y la función inversa. Debajo de estos, se encuentran las funciones para operar con la pila: entrar datos (*enter*), borrar (*DEL*), limpiar la pila (*CLR*), intercambiar las dos últimas posiciones de la pila (*SWAP*) y recuperar la última pila (*LSTK*).

A la izquierda del teclado numérico se encuentran las funciones matemáticas. En la primera fila se encuentran las funciones para operar con números complejos, en la segunda están las teclas para introducir números complejos, pi y la función factorial. El resto de teclas se agrupan en las funciones trigonométricas y sus inversas, y en las exponenciales y logarítmicas.

Finalmente, en la parte superior derecha están las teclas para los modos de pantalla (*NORM*, *FIX* y *SCI*), las unidades angulares (*DEG*, *RAD*, *GRAD*) y el sistema de coordenadas para trabajar con números complejos (*CART* para cartesianas y *POLAR* para polares). La opción seleccionada se destaca visualmente cambiando el color de la letra.

8.4.3 Ayuda

La calculadora cuenta con una ayuda que explica cómo manejarla y realizar cálculos. La ayuda se muestra cuando se pulsa el icono correspondiente y aparece en una ventana independiente que se puede desplazar por la pantalla. Está estructurada como una FAQ con preguntas y respuestas sobre temas concretos acompañadas de múltiples ejemplos realizados con gráficos SVG. La idea es que el acceso a sea claro, rápido y ágil.



Il·lustració 60 Pantalla de ajuda de la calculadora

8.4.4 Funcionamiento

Como ya se ha comentado anteriormente, esta calculadora utiliza el sistema de notación polaca inversa (RPN). Este sistema permite convertir fácilmente una operación con paréntesis en una concatenada que se implementa sin dificultad en cualquier lenguaje de programación. Por el contrario, si se intenta programar la misma operación en

¹⁵⁶ La calculadora utiliza los signos de puntuación del sistema numérico anglosajón; en este caso un punto.

notación normal, hay que programar un analizador de expresiones que al final también acaba utilizando sintaxis RPN para hacer los cálculos.

Respecto a su funcionamiento, la calculadora se divide entre una pila de datos, donde se almacenan los valores introducidos y los calculados, y un gestor de pantalla que muestra la información en pantalla. Responde al paradigma de programación MVC que se utiliza en todo el proyecto.

Para realizar una operación el usuario empieza por introducir los valores. Si no está presente, la línea de comandos, se activa y se van introduciendo los números. El programa analiza la sintaxis para evitar errores y una vez pulsada *enter* se realiza una comprobación final del formato del número y se guarda en la pila. Internamente la calculadora guarda los números como objetos de tipos números real o complejo. Cuando se pulsa una función u operación, el sistema comprueba si está definida para ese tipo¹⁵⁷ y si hay suficientes argumentos para realizarla. Luego extrae los elementos necesarios de la pila, hace las conversiones de tipos necesarias y ejecuta la operación. El resultado se almacena en la pila y el proceso finaliza reescribiendo la pantalla. En caso de error, se mostrará un mensaje en la parte superior de la pantalla.

1:	3
2:	1: 3 5
3:	2: 3 1: 5
2:	1: 8

Ilustración 61 Secuencia de operaciones de 3 + 5

8.4.5 Implementación

Además de los métodos comunes al resto de herramientas de la barra, la calculadora implementa otros necesarios para poder realizar los cálculos. Todo el código se ha realizado según los paradigmas de la programación orientada a objetos (POO) y la arquitectura MVC. En el código, se ha separado la gestión de la pantalla, de la interacción con el usuario y la lógica de negocio.

La lógica de negocio está gestionada por tres objetos: la pila de datos, números reales y números complejos. El primero controla la estructura donde se almacenan los datos y los otros dos se encargan de realizar las operaciones en función del tipo de datos almacenado.

La estructura de datos sobre la que se organiza la calculadora es una pila LIFO (*Last In First Out*). En esta estructura se cumple que el último elemento en entrar en ella es el primero en salir. Es como un *Castell* o torre humana; se descarga en el orden inverso a como se montó, empezando por arriba. Para utilizar una pila, es necesario pues, implementar funciones que permitan crear la pila, añadir elementos, sacarlos, calcular el número de componentes y borrar todo su contenido.

La razón de escoger una estructura LIFO, tiene que ver con la mecánica de las operaciones. Programar una secuencia de operaciones sencillas, como una suma de dos números, es muy fácil. Pero cuando hay que encadenar expresiones, incluir paréntesis, funciones dentro de funciones... y entran en juego las reglas de precedencia de las operaciones, la

¹⁵⁷ Por ejemplo, los números complejos no tienen definida la función factorial.

cosa se vuelve muy complicada. Evaluar una expresión como la siguiente no es complicado para los seres humanos que disponen de una visión global de la operación pero sí para un ordenador que la procesa secuencialmente. De un vistazo, un humano, sabría que primero hay que calcular el argumento del seno, después el seno, a continuación hacer el producto y finalmente las sumas y restas.

```
3+5·sin(25+65)
```

Código 47 Ejemplo de expresión matemática compleja

Pero existe otra manera de escribir esta expresión, y es poner primero los números y después las operaciones. Reformulemos el ejemplo anterior.

```
3 5 25 65 + sin · +
```

Código 48 Expresión matemática en RPN

Esta segunda forma es mucho más fácil de programar que la primera. Como la calculadora no implementará un evaluador de expresiones, los cálculos como el anterior se realizarán aprovechando la capacidad de encadenar operaciones.

A continuación se va a ver con más detalle el ejemplo anterior, en el supuesto de que el modo angular actual sea DEG. En primer lugar se introduce el tres. Después el cinco, el veinticinco y el sesenta y cinco. Hasta ahora, no se ha hecho ninguna operación porque faltan los resultados parciales de las operaciones con mayor precedencia. A continuación, se suman los dos últimos elementos introducidos (25 y 65) y se obtiene el primer resultado parcial, 90. Ahora ya se puede calcular el seno de 90 y obtener uno. El resultado se multiplica por cinco y finalmente se le suma tres.

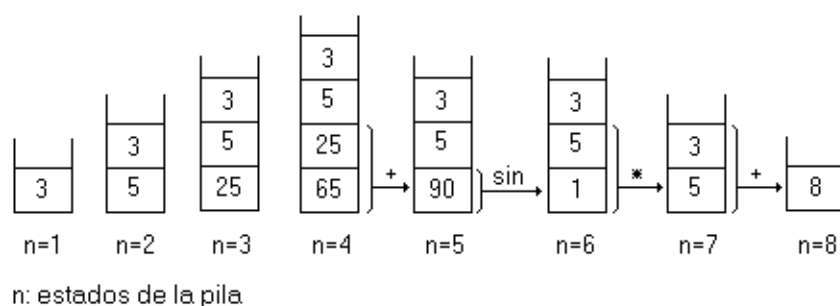


Ilustración 62 Proceso de cálculo de $3+5\cdot\sin(25+65)$

De la figura se extraen valiosas conclusiones. Por un lado, la pila es un elemento dinámico de longitud variable donde se crean y se destruyen componentes a medida que hacen falta o dejan de hacerla. Y por otro, los nuevos datos siempre son los primeros en salir. La idea general es almacenar los números y después ir haciendo las operaciones parciales a medida que se pueda por orden de precedencia.

Implementar una pila LIFO no es difícil en javascript. El propio lenguaje incorpora funciones para tratar los vectores como tales, o el programador puede construir sus propias funciones. En este caso es necesario implementar dos pilas, una para realizar los cálculos y otra a modo de copia de seguridad antes de realizar cualquier operación. De esta manera se puede recuperar el último valor si el usuario se ha equivocado o lo considera pertinente.

El objeto con la pila de datos, incorpora además un analizador sintáctico de números reales y complejos escritos en la línea de comandos, y la gestión de las operaciones matemáticas. El analizador evita, por ejemplo que en un número aparezca más de un separador decimal¹⁵⁸. Para ello recurre a la evaluación de expresiones regulares. Es un método muy

¹⁵⁸ El separador decimal utilizado por esta calculadora es el punto.

potente para validar expresiones, desechado a menudo injustamente por programadores poco experimentados. La gestión de operaciones se encarga de extraer los números para realizar las operaciones, comprobar que se puede realizar la operación y en caso de que los tipos sean diferentes, hacer las conversiones necesarias.

Para trabajar con números, se implementan dos objetos que representan los números complejos y los reales. Cada uno de ellos implementa todas las funciones matemáticas necesarias y las operaciones para hacer las conversiones entre los diferentes tipos. Para implementar las funciones matemáticas se han utilizado las funciones del objeto *Math* de javascript o expresiones algebraicas estándar¹⁵⁹ que resultan de la combinación de funciones simples. Trabajar de esta manera facilita expansiones futuras; por ejemplo, si se quisiera añadir la capacidad de cálculo vectorial, sólo habría que programar el correspondiente objeto. Se puede discutir si no sería mejor utilizar desarrollos de Fourier para hacer los cálculos, pero seguramente, dada la capacidad de cálculo de los dispositivos actuales, no se notaría una gran diferencia.

La interacción con el usuario es una combinación de javascript y SVG. Sobre cada uno de los nodos con las teclas, se ha definido un controlador de eventos que invoca una rutina. La función controla que el número de argumentos pasados sea suficiente y muestra un mensaje en caso de error. En el caso de las teclas numéricas, se cambia al modo de la línea de comandos, si no está en activo y se va verificando la sintaxis del número a medida que se escribe. Con las teclas de los modos de pantalla se comprueba si se ha pasado el número de dígitos (FIX o SCI), o se selecciona directamente en el caso de las unidades y sistema de coordenadas. Si es necesario, se regenera la pantalla.

La pantalla se aprovecha de las capacidades de SVG para incluir texto en las imágenes. El nodo SVG con el texto es manipulado por javascript para generar el modo de pantalla adecuado y escribir en él los valores y mensajes. Una ventaja adicional de operar de esta manera es que el texto se puede seleccionar y copiar para pegarlo en otra aplicación. Cómo se representan los números viene determinado por el modo, las unidades angulares y el sistema de coordenadas seleccionado por el usuario.

La generación de la pantalla es un proceso complejo. Una vez el usuario ha realizado la operación, se invoca el método *imprimirPantalla()* del objeto que controla la pantalla. Este empieza a recorrer la pila por el final y va rellenando las líneas hasta que se acaban o se agotan los elementos de la pila. Para cada uno de ellos, número real o complejo, invoca a un método llamado *toString()* que genera la representación en pantalla. Este consulta el modo de presentación (NORM, FIX, SCI), las unidades angulares y en caso de complejos, el sistema de coordenadas (polar o cartesianas) para determinar el formato del número.

Una de los elementos más satisfactorios de cómo se ha programado la calculadora, es que al ser modular, los objetos creados se pueden aprovechar en otros proyectos. Además, se pueden incorporar en el futuro nuevas funcionalidades como capacidades gráficas, evaluación de expresiones, librerías de fórmulas, resolución de ecuaciones por métodos numéricos, cálculo con matrices, etc. No obstante, se ha considerado que las prestaciones actuales son más que suficientes como complemento de los cursos.

8.5 Herramienta notas

Las notas permiten que el usuario pueda añadir sus propias anotaciones y comentarios e incorporarlos a las explicaciones. De esta manera, el aprendizaje se enriquece y el alumno puede hacer sus propias aportaciones. Se consigue así uno de los objetivos del curso, facilitar herramientas para que el usuario pueda adaptar los contenidos a sus necesidades y que los recursos utilizados no sean meramente estáticos.

¹⁵⁹ Véase por ejemplo las funciones hiperbólicas en https://en.wikipedia.org/wiki/Hyperbolic_function

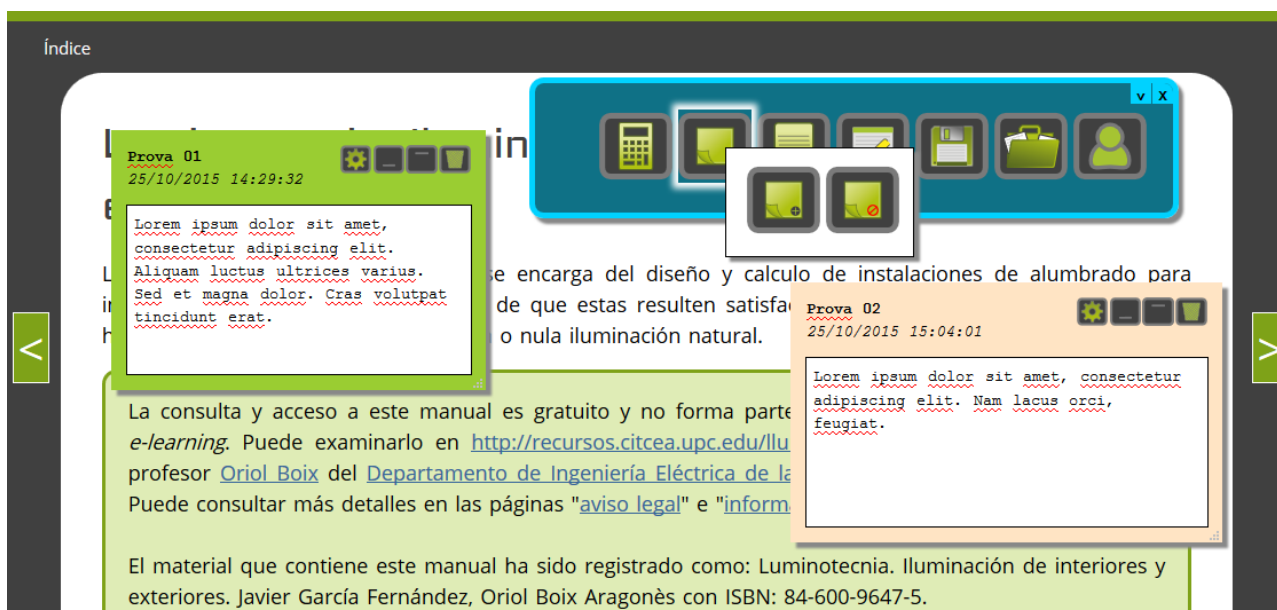


Ilustración 63 Interfaz de usuario de las notas y ejemplo de nota

El hecho de que la interfaz de usuario se asemeje a un *post-it* es algo completamente intencionado. Se busca que visualmente y de manera inmediata el usuario aprecie que es un añadido, un elemento ajeno al contenido original del curso.

En la barra de herramientas está ubicado el icono de la aplicación que permite acceder al menú contextual. En este hay dos opciones disponibles: crear una nueva nota y ocultar/mostrar todas las notas.

La nota consta de la barra del título y una zona para escribir texto. En la barra se sitúan la fecha de creación de la nota y un grupo de iconos que realizan diferentes funciones: minimizar la ventana, maximizarla para recuperar el tamaño original, eliminar la nota y cambiar el color de la ventana. El área de texto es la zona reservada para las anotaciones donde el usuario puede introducir libremente texto sin formato.

Se decidió no dar formato al texto que el usuario introducía en las notas por diferentes motivos: porque resultaba más fácil de implementar, porque reforzaba la idea de que se trataba de un añadido y porque, en esencia, añadir estilo al texto no aportaba beneficios adicionales a la herramienta.

El título de la nota también es editable. El borde de la ventana, en color verde en la imagen, es una zona activa que cuando se pulsa encima permite desplazar la ventana sobre la pantalla. En el caso de la esquina inferior derecha, se puede cambiar el tamaño.

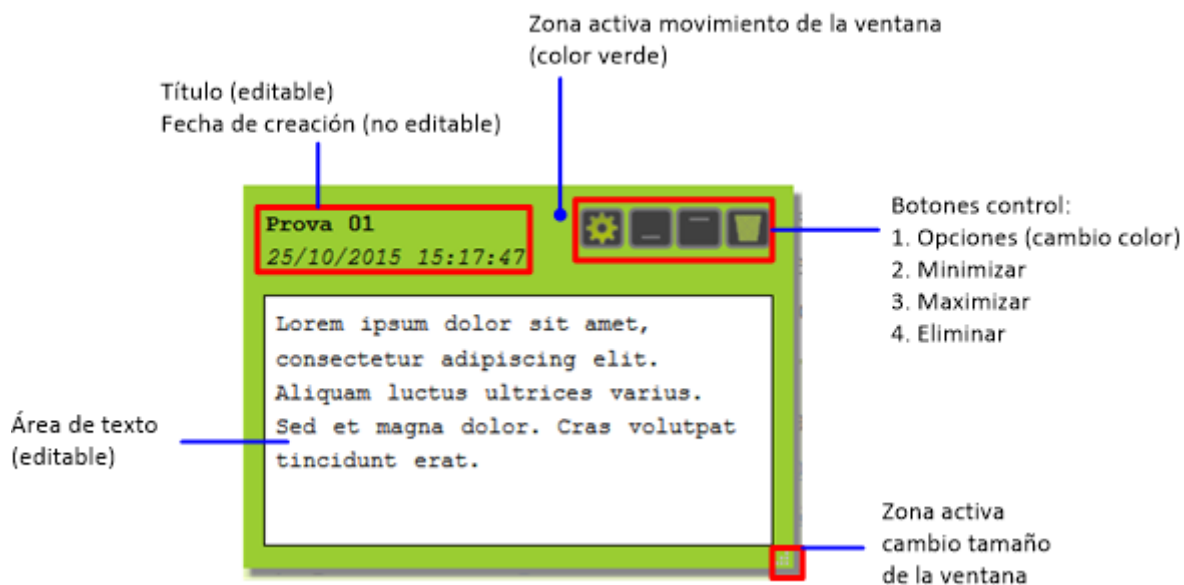


Ilustración 64 Nota

Dado que las notas ocupan bastante espacio en la pantalla y pueden resultar molestas para seguir el curso, se incorporó la posibilidad de ocultarlas, minimizarlas (y quedan reducidas a la barra del título), cambiar el tamaño y desplazarlas por la pantalla. Cuando se mueve la nota, el programa guarda la posición en la pantalla de manera que siempre aparece en el mismo sitio aunque se produzca desplazamiento (*scroll*) en la pantalla. Lo mismo ocurre con el tamaño de la ventana.

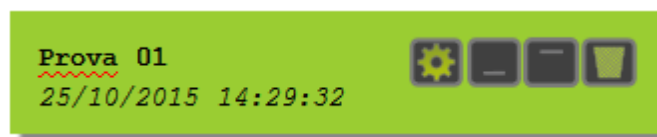


Ilustración 65 Nota minimizada

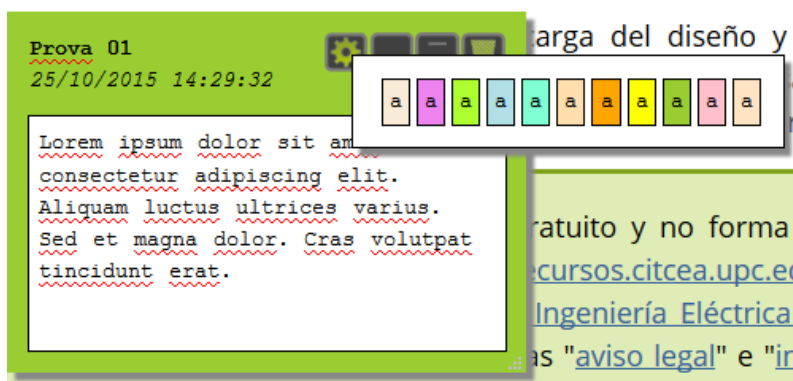


Ilustración 66 Nota con menú para seleccionar el color

Cuando se crea una nueva nota, el color es escogido aleatoriamente dentro de la paleta disponible, aunque el usuario lo puede cambiar a posteriori. Se han utilizado colores chillones con el objetivo de que las notas se destaquen claramente. Se ofrecen varios colores para que el usuario pueda asignarles, libremente, tipologías de nota o bien órdenes de prioridad o importancia según su propio criterio.

A nivel de estructura, la aplicación sigue las grandes líneas de la arquitectura descrita al comienzo del capítulo. Es una aplicación cliente-servidor en la que la información se almacena en la BD.

El funcionamiento también es similar al del resto de herramientas. Cuando se carga la herramienta, el cliente envía al servidor cuál es la página visitada. Este consulta la base de datos para extraer la lista de notas del usuario. Si hay notas, envía la lista en formato JSON al navegador que crea las notas a partir de los datos almacenados. Así, el usuario recupera las notas con las medidas, color, posición y texto que tenían en su última visita a la página.

La implementación fue bastante más compleja que otras herramientas porque fue la primera que se realizó. En el cliente, la interfaz gráfica estaba compuesta por varios elementos HTML, algunos de los cuales tenían que ser editables (el título y el texto), la ventana se tenía que poder desplazar por la pantalla, cambiar el tamaño y el color. Además había que evitar conflictos de eventos que provocaban, por ejemplo, que al pulsar el ratón sobre el área de texto empezara a moverse o que hubiera conflictos al intentar redimensionar la ventana. Al final, poco a poco se pudieron ir solucionando.

La causa principal de los problemas de implementación era el burbujeo de eventos. Este consiste en el conjunto de controladores que se disparan cuando se produce un determinado evento. Las causas de esto hay que buscarlas en que:

- El elemento tiene asociados eventos por defecto (botones, enlaces ...)
- Se ha definido más de un controlador (*listener*) para un mismo evento y elemento.
- El elemento sobre el que se ha producido el evento contenga otros elementos donde se han definidos otros controladores de eventos. Si por ejemplo se define un evento sobre una etiqueta que contiene un enlace <a> se disparan dos eventos, uno por cada etiqueta.
- Que el evento implique disparar otros. Por ejemplo:
 - mousedown -> mouseup -> mouseclick
 - keydown-> keyup -> keypress

Los problemas con los eventos en las notas se solucionaron limitando las zonas donde se podían disparar. Así desde el área de texto y el título no se puede mover la ventana; ni tampoco desde la esquina inferior derecha que se reserva para cambiar el tamaño.

El cambio de tamaño en las notas implicaba problemas con el tamaño del área de texto, ya que debía mantener la distancia con los bordes de la nota y la forma. Otra dificultad estaba relacionada con el botón de minimizar, puesto que no tenía sentido que al pulsarlo la altura ventana fuera inferior a la de la barra del título, en caso contrario no se verían los botones. Tampoco tenía sentido poder minimizar dos veces...

En el área de texto apareció otro problema relacionado con la solución adoptada para que fuera editable. Utilizar un campo <textarea> de un formulario HTML no era una idea muy atractiva por dos razones. Primero, el código HTML y el javascript es más complejo. Y, segundo, dentro de los campos de texto sólo se puede utilizar texto plano; por lo tanto hay que olvidarse de utilizar etiquetas HTML. Esto impedía por ejemplo, poder crear enlaces. Así pues, se optó por añadir un atributo *editable* a una etiqueta de párrafo <p>. El inconveniente está en que para poder utilizar saltos de línea hay que utilizar etiquetas
 pero al guardar los textos en la BD, el filtro de seguridad que elimina etiquetas HTML los suprime. Al final la solución consistió en sustituir los
 por \n¹⁶⁰ antes de enviar las notas al servidor.

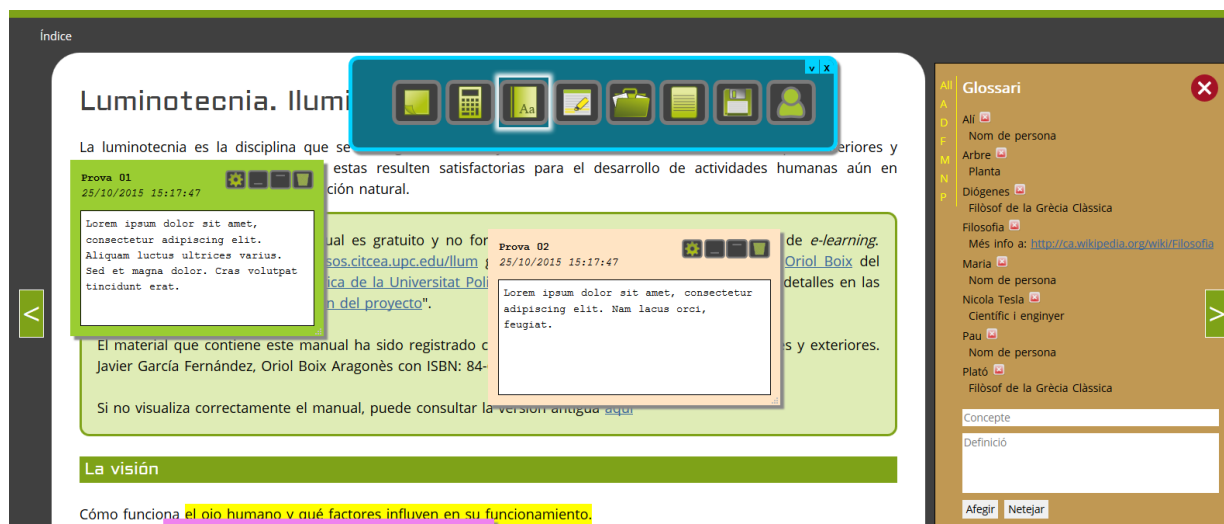
Para almacenar las notas se utiliza un *array* de objetos cada uno de los cuales tiene toda la información necesaria para crear una nota. Estos objetos contienen, además, todas las funciones necesarias para manipular las notas.

En el lado del servidor, el funcionamiento es como en el resto de herramientas. La única particularidad es que en vez de trabajar con un único elemento (resumen, subrayado...) se envían todas las notas de la página de golpe. Estas se almacenan individualmente en una tabla de la BD.

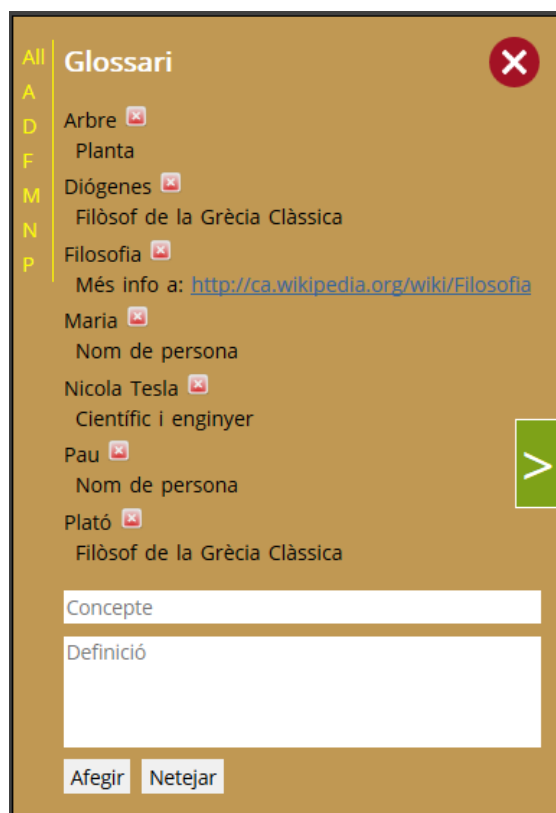
¹⁶⁰ Secuencia de escape que equivale a un salto de línea más retorno de carro.

8.6 Herramienta glosario

El glosario es una herramienta creada para que el usuario pueda realizar su propia lista de términos sobre el curso. Al contrario del subrayado, los resúmenes o las notas, que son exclusivos de cada página, el glosario es único para todo el curso. Es un listado de parejas concepto-definición.



Il·lustració 67 Aspecto del glosario



Il·lustració 68 Interfaz gráfica del glosario

La interfaz gráfica se divide en tres partes. A la izquierda, en dirección vertical, se sitúa el índice con todas las letras para las que hay entrada. Proporciona un acceso rápido al directorio. Al principio, aparece la entrada *A//* desde donde se accede a toda la lista. En la parte superior derecha están el botón de esconder la aplicación y la lista de entradas ordenadas por orden alfabético. Al lado de cada entrada hay un botón para borrarla y si se pone el cursor sobre la

definición ésta puede editarse. No se permite hacer lo mismo con el término para evitar problemas de duplicación de claves en la BD. Finalmente, en la parte inferior existe un formulario para poder introducir nuevas entradas.

Para la implementación del glosario, se optó por mantener la lista de definiciones en un *array* en la memoria del navegador. Las conexiones con el servidor solo se hacen cuando se carga la herramienta, momento en que se descarga la lista de la BD, y al cerrarla, momento en el que se guarda. De esta manera se reduce la conectividad con el servidor y se aumenta la velocidad de la aplicación. Cada uno de los términos de la lista se almacena en un objeto con cuatro propiedades, concepto, definición y dos indicadores; uno para saber si se ha editado el término y otro para saber si está en la BD. El índice del glosario se almacena en otro *array* en el que se guarda el carácter y la posición de la lista en el que está el primer término que empieza por él. Es una lista que se genera dinámicamente a media que se añaden y eliminan elementos de ella.

Respecto a los términos que se pueden incluir en el glosario, no hay restricciones. Pueden comenzar por cualquier carácter y en la definición se pueden incluir también enlaces a páginas web. Cuando se añade el término, un analizador busca mediante expresiones regulares los enlaces y los convierte en etiquetas <a> de HTML. Es algo similar al funcionamiento de los *hashtag*¹⁶¹ de Twitter.

8.7 Herramienta resumen

El resumen es una técnica que permite a los alumnos expresar con sus propias palabras las ideas principales del texto de una forma lineal. Esta aplicación que puede recordar a las notas, parte de una premisa diferente. Mientras que con estas se busca que el usuario pueda añadir comentarios o complementar la información presente en el texto, los resúmenes aportan al usuario la posibilidad de definir y estructurar las ideas principales que aparecen en la página.

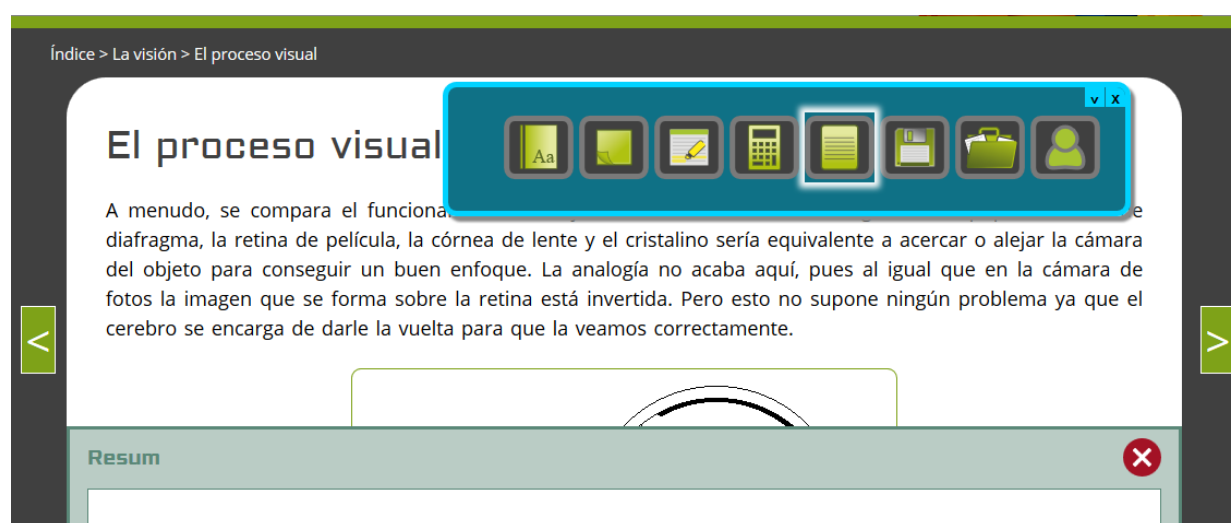


Ilustración 69 Herramienta del resumen

La interfaz de usuario consta de una ventana con el título de "Resumen", un botón para cerrar la ventana y un recuadro blanco donde se puede escribir. La ventana aparece en la parte inferior de la pantalla donde ocupa todo el ancho del área del artículo. Es importante que esté al final para reforzar la idea de que es un resumen. Sin embargo, y para

¹⁶¹ "Una etiqueta o *hashtag* [...] es una cadena de caracteres formada por una o varias palabras concatenadas y precedidas por una almohadilla o numeral (#). Es, por lo tanto, una etiqueta de metadatos precedida de un carácter especial con el fin de que tanto el sistema como el usuario la identifiquen de forma rápida.

Se usa en servicios web tales como Twitter, Telegram, FriendFeed, Facebook, Google+, Instagram, Weibo o en mensajería basada en protocolos IRC para señalar un tema sobre el que gira cierta conversación." [145]

ofrecer una mayor comodidad en su uso, se ofrece la posibilidad de desplazarlo verticalmente. El botón de cerrar oculta el resumen para que no moleste mientras se lee el texto. En cualquier momento se puede volver a ver pulsando sobre el botón de la barra de herramientas que también tiene la función de esconder y mostrar la ventana. En el cuadro de texto, se ha utilizado una nueva característica de HTML5 que permite convertir en editable cualquier etiqueta de texto. De esta manera, se evita utilizar múltiples formularios HTML y se simplifica el código.

Un aspecto importante era diferenciar gráficamente el resumen de las notas, para que no se prestara a confusión. Por este motivo, el ancho y el alto son fijos, no se puede cambiar el color y sólo se puede desplazar verticalmente. Para evitar conflictos de eventos entre el movimiento de la ventana y la escritura en el cuadro de texto (copiar, pegar, cortar...), sólo se puede desplazar la ventana pulsando sobre las zonas en verde.

Los textos de los resúmenes se almacenan en una tabla en la base de datos de la aplicación.

El funcionamiento de la aplicación es simple. Cuando se carga la herramienta, esta solicita al servidor que le devuelva el resumen de la página que el usuario está visitando. Si existe, la BD envía el texto que se muestra en la ventana del resumen. El usuario puede leerlo, editarlo o borrarlo. Cuando se cambia de página o se abandona la sesión, los cambios se guardan en la BD.

La implementación sigue las pautas generales de las herramientas de la barra ya explicadas. El problema más significativo fue cómo ajustar la amplitud de la ventana al mostrar la herramienta glosario. Cuando esta es visible, se produce un cambio en la distribución de los elementos en la pantalla y en el ancho del artículo que se utiliza como referencia para conocer la amplitud de la ventana del resumen. El problema se acentúa porque en javascript no existe un evento para controlar el tamaño de un objeto o los cambios en objetos del DOM. Por ello hubo que recurrir a una nueva característica de javascript, los observadores de mutaciones o *MutationObserver*¹⁶², que permiten definir acciones como consecuencia de cambios en el DOM.



Il·lustració 70 Aspecto del resumen cuando se muestra el glosario

¹⁶² <https://developer.mozilla.org/en-US/docs/Web/API/MutationObserver>
<https://hacks.mozilla.org/2012/05/dom-mutationobserver-reacting-to-dom-changes-without-killing-browser-performance/>

8.8 Herramienta subrayado

El subrayado es una técnica de estudio que consiste en marcar las palabras o partes del texto más relevantes donde se encuentran las ideas principales. El usuario puede así destacarlas y guardarlas en la base de datos para que estén disponibles en sucesivas visitas a la página.

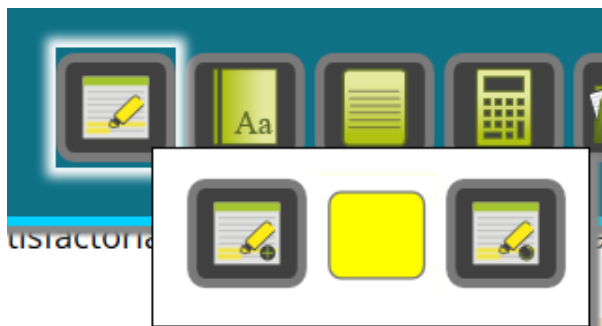


Ilustración 71 Aspecto del menú de la herramienta

Respecto a la interfaz, cuando se pulsa encima del icono posicionado en la barra de herramientas, aparece un menú contextual con tres opciones: la primera permite marcar un texto seleccionado, la segunda escoger un color y la tercera eliminar todos los subrayados.

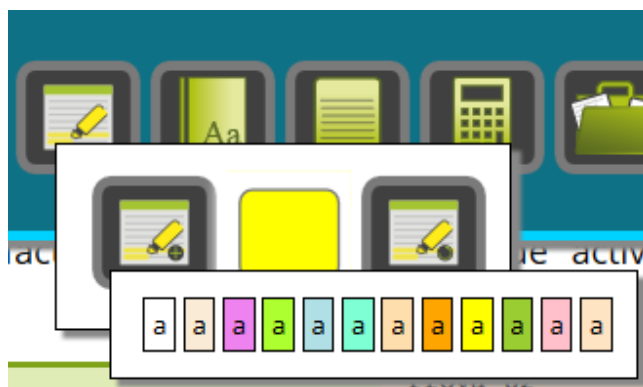


Ilustración 72 Ventana para seleccionar el color de marcado

Para marcar un texto hay que seleccionarlo en primer lugar y a continuación pulsar el primer icono del menú contextual. Si se quiere cambiar el color, hay que hacerlo antes de crear el subrayado. Para cambiar el color de un texto ya marcado, se selecciona y se señala con el nuevo color. Para eliminar un subrayado basta con marcarlo por encima con color blanco tal y como se hace en los paquetes de ofimática.

La implementación de esta herramienta fue especialmente compleja, tanto que hubo que desechar diferentes soluciones, algunas en avanzado proceso de implementación.

Para comprender las dificultades, conviene primero conocer cómo se almacena en la memoria del navegador la página que se visualiza en pantalla. El documento HTML se guarda en la memoria en una estructura con forma de árbol conocida como DOM (*Document Object Model*¹⁶³) donde se relacionan todas las etiquetas. El árbol consta de un único nodo raíz del que cuelga el resto de elementos. Los nodos del árbol tienen, entre ellos, relaciones filiales padre-hijo o de hermanos según su posición en este. No todos los nodos del árbol coinciden con etiquetas HTML; también lo son los comentarios y los textos. A continuación puede verse un ejemplo de documento HTML y su DOM correspondiente.

¹⁶³ Modelo de objetos del documento.

DOM

nivell 2

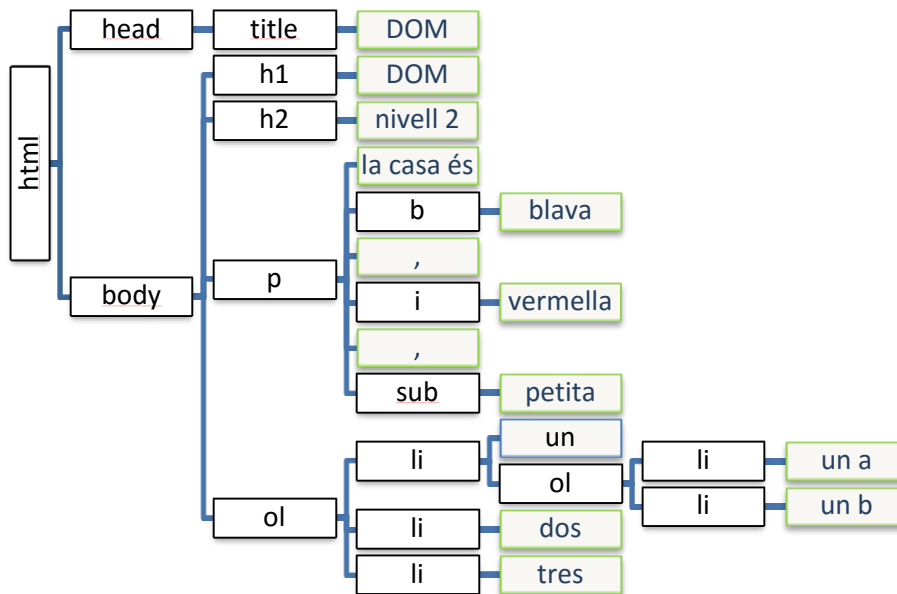
la casa és **blava**, *vermella*, _{petita}

1. un
 1. un a
 2. un b
2. dos
3. tres

Il·lustració 73 Visualització en pantalla

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>DOM</title>
</head>
<body>
<h1>DOM</h1>
<h2>nivell 2</h2>
<p>la casa és <b>blava</b>,
<i>vermella</i>, <sub>petita</sub></p>
<ol>
<li>un
<ol>
<li>un a</li>
<li>un b</li>
</ol>
</li>
<li>dos</li>
<li>tres</li>
</ol>
</body>
</html>
```

Código 49 Código HTML



Il·lustració 74 Àrbol DOM

El primer problema que surgió con la herramienta del subrayado fue el de cómo acceder desde javascript al texto seleccionado y cómo identificar los puntos de inicio y final en sus respectivos nodos. La solución vino de la mano del método `getSelection()`¹⁶⁴ del objeto `window` que identifica el texto seleccionado y lo guarda en un objeto `range`¹⁶⁵. `Range` es un objeto que representa un fragmento de documento que contiene nodos y partes de nodos de texto. A continuación, se va a desarrollar un ejemplo donde se podrá ver la problemática existente.

DOM

nivell 2

la casa és blava, vermella, petita

1. un
 1. un a
 2. un b
2. dos
3. tres

Ilustración 75 Ejemplo de texto seleccionado

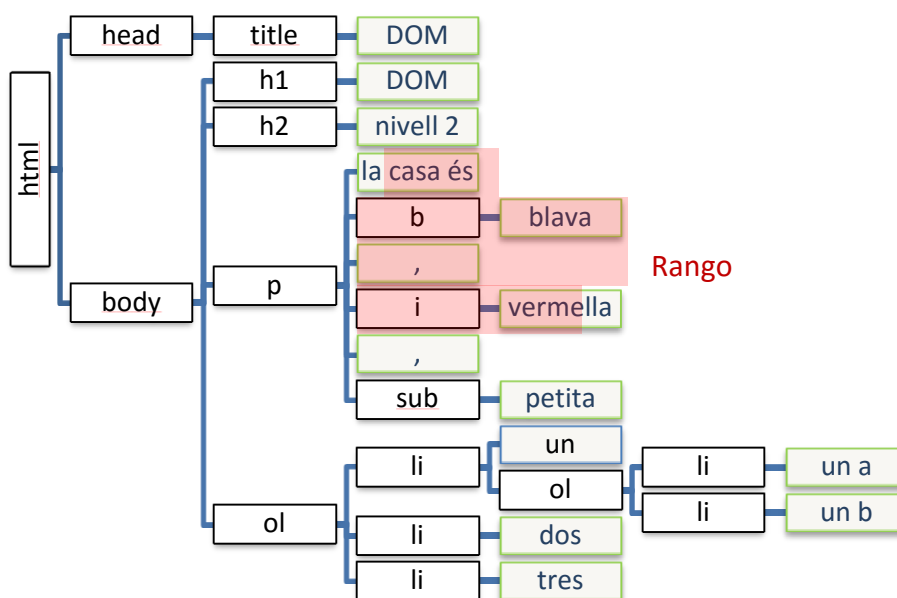


Ilustración 76 DOM con rango seleccionado

En este ejemplo, el rango seleccionado comienza en el nodo de texto "*la casa és*" al que le sigue un nodo `` que incluye el nodo de texto "*blava*", luego otro nodo de texto con "," y finalmente un nodo `<i>` donde se incluye el nodo de texto con la parte final seleccionada. Como se puede ver, el rango incluye partes de nodos de texto y nodos con etiquetas HTML.

Por otro lado, para subrayar o marcar un texto hay que incluirlo dentro de una etiqueta y aplicarle un estilo. Esto supone alterar el DOM. Existen dos tipos de casos posibles:

¹⁶⁴ <https://developer.mozilla.org/en-US/docs/Web/API/Window/getSelection>

¹⁶⁵ <https://developer.mozilla.org/en-US/docs/Web/API/Range>

1. Todo el texto seleccionado está dentro de un único nodo de texto. En este caso, hay que sustituir el subárbol que cuelga de <p> por otro con varios nodos (tres en este caso), dos de texto a derecha e izquierda, y uno con una etiqueta , que se utiliza para aplicar el estilo, del que cuelga a su vez el texto marcado. Puede verse que incluso en este caso, el más sencillo, la complejidad es considerable.

```
<p>la casa és blava, vermella, petita</p>  
se transforma en  
<p>la <span>casa és blava, verme</span>lla, petita</p>
```

Código 50 Marcado en un nodo de texto único

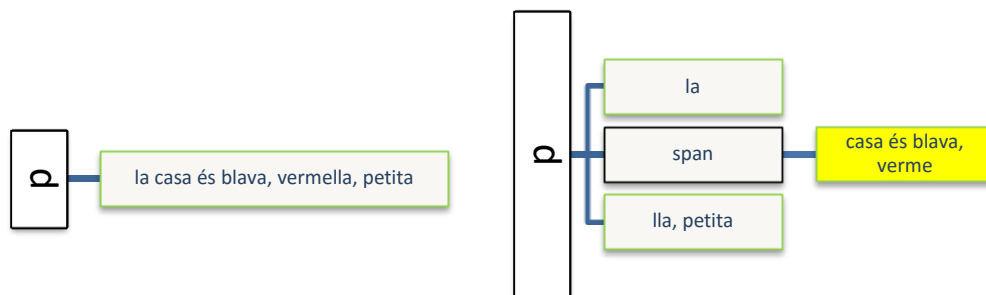


Ilustración 77 Transformación del DOM para un texto sencillo

2. El texto seleccionado está repartido en varios nodos. En este caso hay que aplicar la mecánica anterior a cada nodo que interviene.

```
<p>la casa és <b>blava</b>, <i>verme</i>lla, <sub>petita</sub></p>  
se transforma en  
<p>la <span>casa és </span><span><b>blava</b></span><span>,</span> </span><i><span>verme</span></i>lla</i>,<br><sub>petita</sub></sub></p>
```

Código 51 Marcado en un texto que ocupa diferentes nodos

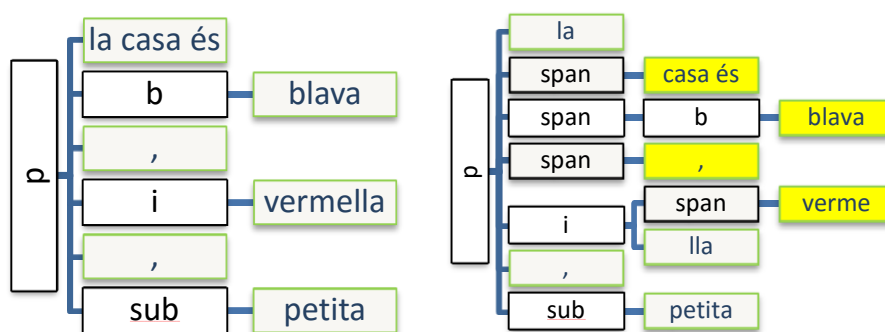


Ilustración 78 Transformación del DOM en un texto que se divide entre varias etiquetas HTML

El resultado final en ambos casos es un fraccionamiento de la estructura del DOM, que se incrementa a medida que se crean más subrayados. Para empeorar aún más las cosas, cuando se eliminan las etiquetas con el marcado, los nodos no vuelven a su estado original. Afortunadamente, existen funciones que reducen el número de nodos de texto del DOM uniendo nodos hermanos contiguos.

La conclusión final es que los efectos de la herramienta de marcado sobre el DOM es la producción de mutaciones en el mismo. Esto conduce a un segundo problema aún más grave: si la estructura del DOM es cambiante, ¿cómo podemos guardar y recuperar la información de los nodos en la BD?

Lo ideal sería guardar sólo los nodos inicial y final del marcado y sus respectivos caracteres de inicio y final; es decir, el rango...Pero, ¿cómo hacerlo si estos cambian a medida que realizamos nuevos marcados? Y es más; en las BD relacionales no se pueden guardar objetos, sólo texto. Por tanto, aunque se guardara el código HTML del fragmento modificado, ¿cómo ligarlo con el HTML original de la página que se guarda en el servidor? Se podría intentar buscar el texto en el documento del servidor, pero es absurdo pues no se puede establecer una relación unívoca (podría aparecer repetido más de una vez o no aparecer si se ha cambiado el contenido).

Tras muchas horas de trabajo se decidió cambiar el planteamiento y pensar primero en cómo resolver el problema de guardar la información en la BD y luego en cómo solventar el tema del marcado.

La solución a este nuevo planteamiento pasa por almacenar todo el código HTML del artículo principal con la explicación de la página en la BD. De esta manera no es necesario buscar nodos y reemplazarlos. Como las páginas, en general, son cortas y no tienen mucho texto, el coste en espacio de almacenamiento se consideró asumible dadas las ventajas técnicas aportadas por la solución. Así, cuando se carga la página, la herramienta solicita a la BD el texto con los subrayados. En caso de que existan, sustituye el código HTML del artículo por el que hay presente en la BD.

Y aun así, resta un último problema, si se modifica el contenido de la página en el servidor, entonces el contenido HTML de la página no coincidirá con el de la BD. Es un problema sin solución; por ello se ha añadido un botón para recuperar el contenido original de la página. En el futuro, se complementará la aplicación con un mensaje de aviso al usuario cuando se produzca esta incidencia.

8.9 Herramienta portafolio

La idea que subyace al portafolio es que el usuario pueda disponer en un único lugar de todos los contenidos que ha creado para enriquecer y adaptar el curso a sus necesidades. De esta manera, el usuario puede hacer un seguimiento de sus progresos y asimilar mejor la materia estudiada. Así se favorece que el trabajo realizado se pueda incorporar a un documento, imprimirlo o compartirlo con otros alumnos y hacer el proceso de aprendizaje más colaborativo.

El portafolio es, por lo tanto, una herramienta modesta que pretende dos cosas:

- Ser un resumen de la actividad del usuario
- Ser en un futuro próximo una herramienta colaborativa, dado que en la versión actual aún no está implementada esta capacidad

No todas las herramientas de la barra se reflejan en el portafolio. Sólo aquellas que permiten desarrollar contenidos al usuario: el glosario, las notas y el resumen. Se podría echar en falta el subrayado, pero no tiene sentido volver a mostrar todo el texto de las páginas; para eso ya está el propio curso.

La interfaz del portafolio está formada por una ventana con una pestaña para cada una de las herramientas que aportan contenido y un botón para cerrarla. En la pestaña del glosario se muestra la lista completa de términos con sus definiciones por orden alfabético, las notas aparecen agrupadas por la página a la que pertenecen, igual que los resúmenes.

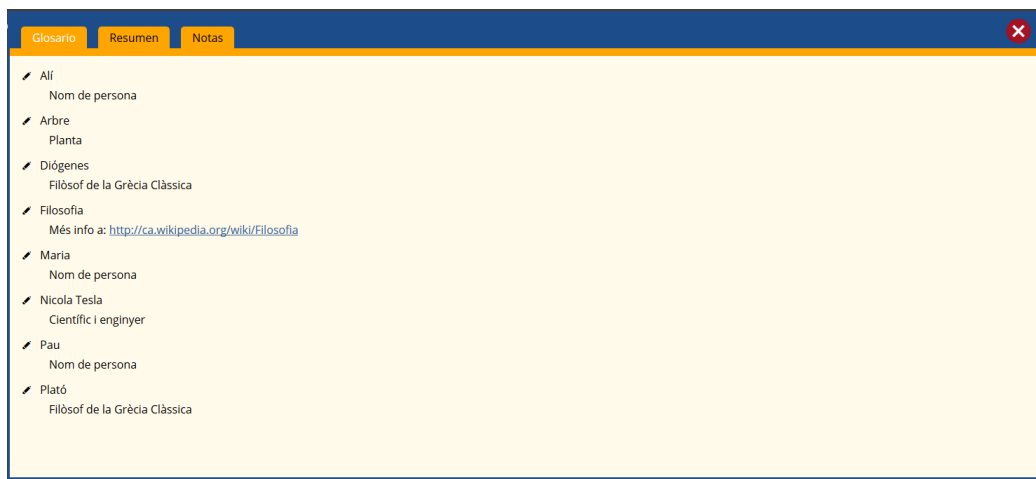


Ilustración 79 Aspecto del glosario en el portfolio

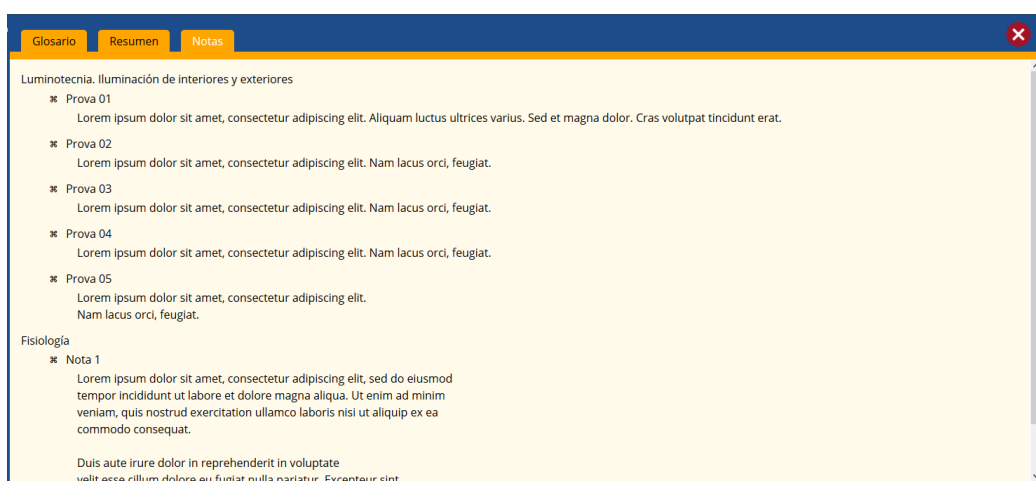


Ilustración 80 Aspecto de las notas en el portfolio

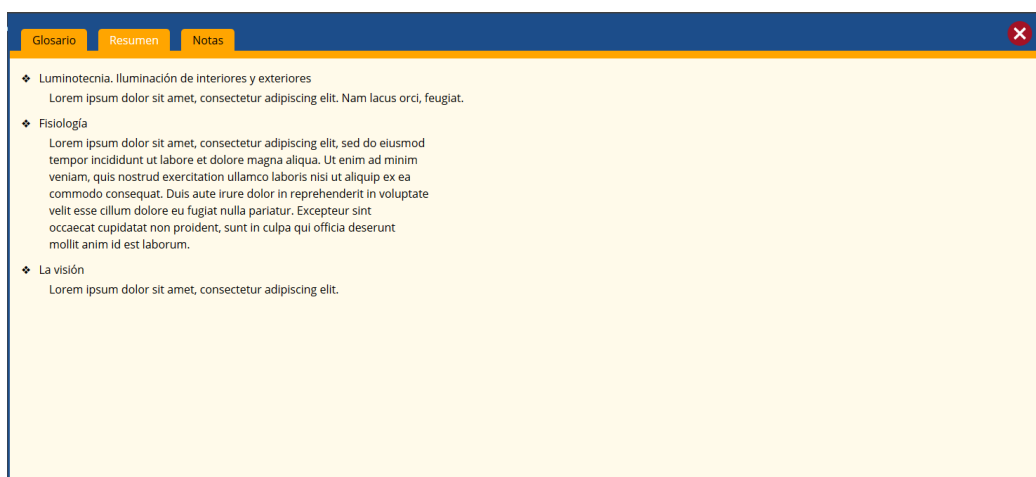


Ilustración 81 Aspecto del resumen en el portfolio

El funcionamiento obedece a la misma arquitectura cliente-servidor del resto aunque con algunas diferencias. En primer lugar, el inicio de la aplicación se retrasa con un temporizador para intentar "asegurar" que las herramientas que generan contenido están cargadas en memoria. Es necesario actuar así porque el portfolio llama a un método, *getPortfolioData()*, presente en las otras aplicaciones que suministra la información para generar las pestañas. Como ya se comentó al hablar de la [barra de herramientas](#), la manera en que están implementados los módulos facilita su

comportamiento como unidades independientes pero tiene aparejado un comportamiento extraño: el orden de carga de los módulos es aleatorio. Por tanto se podría dar la paradoja de que se cargue antes el portafolio y no se creen todas las pestañas porque no están cargados sus respectivos módulos. Aunque no es una solución muy elegante, y es una cosa a mejorar, es la única de las probadas que se demostró como viable. Es un problema pues el retardo sólo se puede determinar experimentando y depende de la velocidad de transmisión de datos en la red. Pero es inevitable si se pretende que las herramientas sean módulos autónomos que se programen de forma independiente.

Tras crear la ventana, que se mantiene oculta hasta que se pulsa en icono de la barra, se cargan los datos de las pestañas. Para ello, se recorre el *array* con la lista de módulos (variable de la barra), se comprueba si generan portafolio¹⁶⁶ y si es así, se llama al método *getPortfolioData()*. Este consulta la base de datos y crea el código HTML con la información. Al final de esta función, se invoca desde el módulo al método *addContent()* del portafolio para generar la pestaña y registrar los eventos del cambio de pestañas. Otra función se encarga de cambiar la pestaña cuando se pulsa encima.

En versiones futuras, la idea es añadir nuevas funcionalidades para imprimir el texto o enviarlo por correo.

¹⁶⁶ En javascript basta con llamar a la función dentro de una estructura condicional para saber si existe. Esto es así porque las funciones definidas dentro de otra función se tratan como propiedades de un objeto.

9 Conclusiones

Una de las contribuciones más importantes que se ha conseguido con este trabajo, es mostrar que los cursos en línea elaborados con HTML pueden ser algo más que un recurso estático y desarrollarse como una herramienta de estudio interactiva en que los estudiantes sean los protagonistas.

Se han trasladado a un soporte electrónico, en Internet y con las tecnologías existentes, algunas de las técnicas de estudio tradicionales. Así, se pueden utilizar notas, crear resúmenes, glosarios o emplear una calculadora para hacer los ejercicios. Los alumnos pueden adaptar el curso a sus necesidades o capacidades, enriqueciéndolo con contenidos creados por ellos mismos. El usuario dispone, por tanto, de herramientas que le convierten en parte activa del proceso de aprendizaje y no en un mero espectador o lector del curso. Con esto, el alumno refuerza y/o desarrolla su capacidad crítica, reflexiva, creativa y de síntesis.

El alumno ampliará sus competencias en TIC, al incorporar los medios tecnológicos a su quehacer diario de estudio tradicional. Sustituye el papel, la calculadora y el diccionario por un soporte informático que le ofrece un “todo en uno” al que puede acceder siempre y donde quiera.

Queda probado que con la tecnología actual no es necesario el uso de sistemas propietarios para crear aplicaciones en el ámbito de la formación en entornos de Internet. Se mantiene, eso sí, la necesidad de disponer de técnicos especializados que programen estas aplicaciones. Las herramientas se han implementado con las tecnologías estándar más comunes existentes en la web. Funcionan sobre navegadores web que cumplen los estándares del W3C y no se han utilizado complementos de terceros. El servidor se ha implementado utilizando PHP, uno de los lenguajes de servidor de aplicaciones más popular y MySQL¹⁶⁷, una de las bases de datos más utilizadas en proyectos web en la actualidad.

Otro elemento destacable es la plataforma de publicación que permite producir y publicar contenidos de una forma simplificada. De esta manera el redactor se puede centrar en preparar los contenidos de los cursos. Añadiendo *scripts* especializados puede incorporar fórmulas o elementos interactivos como ventanas, ejercicios, etc. El aspecto del curso se determina mediante una plantilla y CSS que se pueden modificar fácilmente.

Así mismo, se ha definido e implementado una arquitectura que permite crear las herramientas del estudiante e incluirlas en una aplicación, la barra de herramientas, que incluye la gestión de los usuarios. Este programa es independiente de los cursos y se puede añadir a cualquier página web implementada con HTML. Esto es muy importante porque se puede distribuir de forma separada.

Como punto negativo, se puede mencionar que no se ha podido probar la aplicación con usuarios por falta de tiempo para evaluar sus reacciones e impresiones sobre la barra de herramientas. Hubiera sido muy interesante obtener retroalimentación de los usuarios para mejorar las herramientas. De todas maneras, la implementación de las herramientas que forman este proyecto ya es de por sí un éxito.

Una de las conclusiones más importantes de este estudio, es adquirir conciencia de que en Internet los límites los ponen las personas que desarrollan las aplicaciones. Se pueden crear cursos en línea que se adapten a la forma de aprender del estudiante. No hay que conformarse con crear libros de texto en línea.

¹⁶⁷ En lugar de MySQL, se podría haber utilizado MariaDB; una BD derivada de la anterior pero totalmente libre. <https://mariadb.org/>

10 Desarrollos futuros

De cara al futuro sería interesante ampliar el trabajo en las siguientes áreas:

En primer lugar, habría que avanzar en el desarrollo de la plataforma de publicación. Por un lado, podría facilitarse la capacidad de publicación con la creación de un sistema de gestión de cursos. Desde él se podría crear y eliminar cursos, añadir páginas, editarlas, borrarlas y definir la navegación entre páginas a través de la tabla de contenidos. También se podrían gestionar las plantillas para la maquetación de los contenidos, las hojas de estilo o los archivos de javascript para los recursos interactivos. No se trata de crear un campus virtual tipo Moodle¹⁶⁸, sino más bien un CMS (*Content Management System*)¹⁶⁹ tipo Joomla¹⁷⁰ o Drupal¹⁷¹. Esto implica crear un sistema de gestión de usuarios y permisos para publicar; pero se podría aprovechar gran parte del trabajo realizado para la barra de herramientas pues, en esencia, es igual. Otra idea interesante es que, además, se pudieran editar las páginas desde el propio gestor con alguna librería de javascript¹⁷².

Para mejorar la plataforma, se podría desarrollar una segunda línea de trabajo que se ocupara de la creación de nuevas librerías para crear aplicaciones interactivas para los cursos que permitan crear otros tipos de ejercicios, animaciones, explicaciones, gráficos interactivos con SVG, etc.

En segundo lugar, modificar las herramientas del alumno o programar otras nuevas que faciliten el aprendizaje colaborativo entre los usuarios. De esta manera, los usuarios podrían compartir los conocimientos y contenidos elaborados durante el proceso de aprendizaje con el resto de personas. Esta colaboración, contribuiría a que el alumno aprovechara la experiencia previa de sus compañeros, puedan resolver entre todos las dudas que tengan...

Continuando con el símil del punto anterior, en el que las herramientas tradicionales se incorporan al nuevo quehacer del estudio tecnológico, obtendríamos como mínimo tres nuevas herramientas: compartir apuntes, redactar apuntes en grupo, y aclarar o compartir dudas sobre el estudio con tus compañeros de clase.

Experiencias de este tipo se encuentran en páginas como la que se ofrece en la documentación de PHP¹⁷³ al final de cada página del manual donde se encuentran las notas del usuario. Cualquiera puede añadir comentarios que enriquezcan el contenido con aclaraciones, consejos, problemas... fruto de su experiencia personal. También se puede mencionar el servicio *Yahoo Answers*¹⁷⁴ (respuestas) o *stack overflow*¹⁷⁵ donde unos usuarios plantean sus dudas y preguntas, y el resto de la comunidad les ofrecen sus respuestas.

Cabe destacar un problema inherente a este tipo de experiencias colaborativas: el de la calidad de las aportaciones de los usuarios. Todo el mundo en su etapa de estudiante sabía a quién valía la pena pedir ayuda o quién escribía los mejores apuntes si los tuyos no eran muy buenos o incompletos. Trasladado a Internet, es un problema de prestigio de los redactores. Sin una voz experta que lo supervise, hay que asumir este riesgo. Pero en la sociedad del conocimiento en la que nos encontramos inmersos, una de las primeras cosas que han de aprender los alumnos es a discriminar las fuentes y distinguir la calidad de los contenidos. Aunque no sean expertos en el tema e incluso neófitos, han de

¹⁶⁸ <https://moodle.org/>

¹⁶⁹ "Un sistema de gestión de contenidos [...] es un programa informático que permite crear una estructura de soporte (*framework*) para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás usuarios. [...] Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio web. El sistema permite manejar de manera independiente el contenido y el diseño. [...]" [69]

¹⁷⁰ <https://www.joomla.org/>

¹⁷¹ <https://www.drupal.org/>

¹⁷² Por ejemplo CKEditor <http://ckeditor.com/> o TinyMCE <http://www.tinymce.com/>

¹⁷³ Véase por ejemplo <https://secure.php.net/manual/es/language.basic-syntax.php> las notas de usuario al final de la página.

¹⁷⁴ Página de consultas sobre cualquier tema. <https://es.answers.yahoo.com/>

¹⁷⁵ Página especializada en consulta sobre programación. <http://stackoverflow.com/>

desarrollar un sentido crítico que les impulse a buscar diferentes fuentes y discernir cuales son las más fiables¹⁷⁶. También han de aprender a distinguir las fuentes, no es lo mismo un material que provenga de una revista especializada o del ámbito universitario que el redactado por alguien anónimo. Certezas absolutas no las encontrarán en ninguna parte, pero por lo menos sabrán a qué atenerse.

Dentro de esta línea, se podría incorporar en el futuro una aplicación para que los usuarios pudieran exportar e importar las notas, resúmenes y el glosario de otros compañeros; además de la función ya existente de enviarlas por correo con el portafolio. Incorporar notas de usuario a las páginas donde todo el mundo pudiera escribir también sería una idea interesante. Otras herramientas podrían ser abrir chats o crear grupos de usuarios. Los foros, aunque la idea resulte muy atractiva, en la práctica no se utilizan mucho y por ello sería descartable.

En tercer lugar, sería interesante que se pudiera realizar un seguimiento de la actividad del alumno de cara a comprobar sus progresos. Lo ideal sería que dispusieran de un tutor que les monitorizara y asistiera durante el proceso de aprendizaje, o en su ausencia, que el programa dispusiera de una herramienta automatizada que mostrara al alumno sus avances. En la actualidad el único soporte se realiza mediante correo electrónico y se limita a responder consultas genéricas.

Implementar un sistema de seguimiento de la actividad del alumno y pruebas de evaluación que demostraran la asimilación de conceptos sería bueno para poder utilizar estos cursos como complemento o parte de asignaturas semipresenciales. Una solución sería diseñar un portafolio de actividades que sirviera para demostrar los progresos del alumno y los conocimientos obtenidos.

Estas demandas se podrían resolver diseñando y creando un conjunto de herramientas de soporte que permitan tanto a usuarios como a hipotéticos tutores realizar un seguimiento de los avances de los alumnos. No sería exactamente la misma herramienta para ambos colectivos, tendría algunas diferencias. Para los tutores lo más importante es ver el progreso del alumno: el tiempo efectivo de conexión, los registros de actividades del alumno (sería muy interesante disponer de un histórico del uso de las herramientas por parte de los alumnos), las calificaciones obtenidas en las pruebas de autoevaluación, los ejercicios del alumno, etcétera. Para el usuario, hay que poner más énfasis en que sea consciente del progreso realizado y lo que ha aprendido. Es conocido que en cualquier formación, y más en la que es a distancia, la falta de percepción de utilidad e interés para el alumno es la antesala de la desmotivación y el abandono del curso.

El cuarto desafío, sería profundizar en el tema de la accesibilidad desde un triple punto de vista:

- adaptación al dispositivo
- eliminación de barreras a las personas con discapacidad
- adaptar/ crear herramientas para alumnos con dificultades de aprendizaje

Aunque todos los dispositivos desde los que podemos conectarnos hoy en día a Internet son capaces de visualizar páginas HTML con CSS y javascript, hay dificultades para conseguir que se pueda acceder en igualdad de condiciones desde cualquiera de ellos. Los problemas van desde el tamaño y la resolución de las pantallas a las limitaciones de las capacidades del navegador. Otro problema importante es el del uso de los eventos. Hay que pensar que en las pantallas táctiles, los eventos que activan javascript no son iguales a los de una máquina de escritorio. En el primer caso, son en su mayoría táctiles y predomina el evento de tocar la pantalla (*touch*) que no acaba de ser igual a un clic de ratón, característico de un equipo de sobremesa. Baste notar que mientras el ratón no desaparece nunca de la pantalla, el cursor de una pantalla táctil desaparece cuando se separa el dedo de esta y que las pantallas también responden a eventos complejos como los que se hacen con varios dedos¹⁷⁷ [48], [49], [50].

¹⁷⁶ En casos de páginas de resolución de dudas sobre programación, si no se es capaz de discernir la mejor solución, es conveniente hacer pruebas con los códigos propuestos para comprobar la validez de las respuestas.

¹⁷⁷ Por ejemplo ampliar-reducir la pantalla al separar-juntar dos dedos.

El proyecto objeto de esta tesis se ha implementado sobre un navegador de escritorio y necesita algo de trabajo extra (básicamente implementar los controladores de eventos) para que funcione en todo tipo de dispositivos.

La eliminación de barreras a la discapacidad es un tema que ha formado parte desde el principio de los objetivos de este proyecto. Se han tenido en cuenta los principios de WCAG y ARIA en la implementación de todas las herramientas, pero convendría un trabajo de refinamiento y mejora continua con pruebas que validen el correcto funcionamiento.

Los alumnos con dificultades de aprendizaje son un colectivo que se podría asimilar con la eliminación de barreras de aprendizaje, pero hay que tener en cuenta que no tienen por qué ser personas discapacitadas.

En colaboración con pedagogos y psicólogos expertos en la materia, se pueden crear herramientas específicas para que alumnos con diferentes necesidades educativas, tengan la oportunidad de mejorar o modificar actitudes de estudio o aprendizaje. Hay usuarios que padecen trastornos de aprendizaje y sólo necesitan que la información se muestre de otra manera o disponer de herramientas adaptadas a su manera de aprender. Es de sobras conocido que hay diferentes formas de aprender y que hay alumnos que requieren de un aprendizaje más práctico, otros de uno más teórico, más colaborativo, individual, etc. Indagar en cómo identificar el estilo de aprendizaje¹⁷⁸ y encontrar la manera de facilitarlo en casos especiales sería una buena manera de contribuir al campo de la formación.

Las herramientas del estudiante han sido diseñadas para un colectivo muy específico, estudiantes de estudios superiores o universitarios. Una vez más, la ayuda de pedagogos, psicólogos y profesores en la creación o modificación de nuevas herramientas ayudarán a un desarrollo futuro del programa, para que éste pueda ser utilizado por alumnos que cursen estudios de ESO o de grado. Se potenciará así la asimilación de las competencias tecnológicas en este colectivo de estudiantes.

¹⁷⁸ [109], [156]

11 Bibliografía

1. *Aprender a aprender: clave para el aprendizaje a lo largo de la vida*. **Martín-Ortega, Elena**. 9, Madrid : Consejo escolar del estado español, 11 de 2008, Participación educativa. Revista cuatrimestral del Consejo Escolar del Estado. Aprender a lo largo de la vida, págs. 72-78. https://sede.educacion.gob.es/publiventa/descarga.action?f_codigo_agc=14202_19. ISSN 1886-5097.
2. **Teorías del aprendizaje**. Constructivismo. *Teorías del aprendizaje*. [En línea] [Data: 05 / 10 / 2015.] <http://uotic-grupo6.wikispaces.com/Constructivismo>.
3. **Wikipedia**. Constructivismo (pedagogía). *Wikipedia*. [En línea] 01 de 10 de 2015. [Citado el: 05 de 10 de 2015.] https://es.wikipedia.org/wiki/Constructivismo_%28pedagog%C3%ADa%29.
4. *El modelo constructivista con las nuevas tecnologías, aplicado en el proceso de aprendizaje*. **Hernández-Requena, Stefany Raquel**. 2, 2008, Revista de Universidad y Sociedad del Conocimiento (RUSC). Universities and Knowledge Society Journal, Vol. 5. <http://dialnet.unirioja.es/descarga/articulo/2799725.pdf>. ISSN-e 1698-580X.
5. **Teorías del aprendizaje**. Conectivismo. *Teorías del aprendizaje*. [En línea] [Citado el: 05 de 10 de 2015.] <http://uotic-grupo6.wikispaces.com/Conectivismo>.
6. **Siemens, George**. Conectivismo: Una teoría de aprendizaje para la era. [En línea] 12 de 12 de 2004. [Citado el: 05 de 10 de 2015.] Traducción: Diego E. Leal Fonseca. http://apliedu.xtec.cat/wiki/form/wikiexport/_media/cursos/tic/d006/modul_1/conectivismo.pdf.
7. E-learning, conectivismo y educación online con Stephen Downes. [En línea] [Data: 02 / 10 / 2015.] <http://geeksroom.com/2010/08/e-learning-conectivismo-educacion-online-con-stephen-downes/30106/>.
8. **W3C**. W3C. *HTML Tags*. [Online] 11 03, 1992. [Cited: 09 26, 2015.] <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>.
9. **Moodle**. Filosofía de Moodle. *Moodle*. [En línea] 6 / 11 / 2013. [Data: 26 / 09 / 2015.] <https://docs.moodle.org/2x/ca/Filosofia>.
10. **Intef**. ¿Qué es un MOOC? *Instituto nacional de tecnologías educativas y de formación del profesorado*. [En línea] [Citado el: 26 de 09 de 2015.] <http://educalab.es/mooc>.
11. **Wikipedia**. MOOC. *Wikipedia*. [En línea] 24 de 09 de 2015. [Citado el: 26 de 09 de 2015.] <https://es.wikipedia.org/wiki/Mooc>.
12. —. Massive open online course. *Wikipedia*. [Online] 09 24, 2015. [Cited: 09 26, 2015.] https://en.wikipedia.org/wiki/Massive_open_online_course.
13. —. M-learning. *Wikipedia*. [Online] 09 20, 2015. [Cited: 09 27, 2015.] <https://en.wikipedia.org/wiki/M-learning>.
14. *Methods and techniques of adaptive hypermedia*. **Brusilovsky, Peter**. 2-3, 1996, User Modeling and User Adapted Interaction, Vol. 6, p. 42. Special issue on adaptive hypertext and hypermedia.
15. **Wikipedia**. Adaptive hypermedia. *Wikipedia*. [En línea] 19 / 07 / 2015. [Data: 29 / 09 / 2015.] https://en.wikipedia.org/wiki/Adaptive_hypermedia.
16. —. Internet protocol suite. *Wikipedia*. [Online] 07 09, 2015. [Cited: 07 22, 2015.] https://en.wikipedia.org/wiki/Internet_protocol_suite.

17. **The Internet Engineering Task Force (IETF).** *RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1*. [Online] June 1999. [Cited: diciembre 04, 2013.] <http://www.ietf.org/rfc/rfc2616.txt>.
18. **Lawton, Shawn Henry, et al.** Developing a Web Accessibility Business Case for Your Organization: Overview. *Web Accessibility Initiative (WAI)*. [Online] W3C, 09 07, 2012. [Cited: 08 05, 2015.] <http://www.w3.org/WAI/bcase/Overview>.
19. **Google.** Guía para principiantes sobre optimización para motores de búsqueda. [En línea] 2011. [Citado el: 04 de 08 de 2015.] https://static.googleusercontent.com/media/www.google.com/ca//intl/es/webmasters/docs/guia_optimizacion_motores_busqueda.pdf.
20. —. Directrices para webmasters. *Herramientas para webmasters de Google*. [En línea] Google, 17 / mayo / 2013. [Data: 01 / julio / 2013.] <https://support.google.com/webmasters/answer/35769?hl=es>.
21. **Fishkin, Rand i MOZ.** The Beginner's Guide to SEO. [En línea] [Data: 05 / 08 / 2015.] <https://moz.com/beginners-guide-to-seo>.
22. **Wikipedia.** Unobtrusive JavaScript. *Wikipedia*. [Online] 03 05, 2015. [Cited: 08 05, 2015.] https://en.wikipedia.org/wiki/Unobtrusive_JavaScript.
23. **Heilmann, Christian.** The seven rules of Unobtrusive JavaScript. [En línea] 2007. [Data: 05 / 08 / 2015.] <http://icant.co.uk/articles/seven-rules-of-unobtrusive-javascript/>.
24. **The Web Standards Project.** The JavaScript Manifesto. *The Web Standards Project*. [En línea] [Data: 05 / 08 / 2015.] <http://www.webstandards.org/action/dstf/manifesto/>.
25. **Ministerio de Educación, Cultura y Deporte.** La propiedad intelectual. *Ministerio de Educación, Cultura y Deporte*. [En línea] [Data: 11 / 08 / 2015.] <http://www.mecd.gob.es/cultura-mecd/areas-cultura/propiedadintelectual/la-propiedad-intelectual/definicion.html>.
26. **Hernandez-Pino, Ulises.** El Derecho de Autor en la Era Digital. *Qué es una Licencia de Uso*. [En línea] 28 / 05 / 2012. [Data: 12 / 08 / 2015.] http://www.iered.org/miembros/ulises/representacion-ideas/Derechos-Autor/qu_es_una_licencia_de_uso.html.
27. **Fundación CopyLeft.** Copyleft. *Fundación CopyLeft*. [En línea] [Citado el: 15 de 08 de 2015.] <http://fundacioncopyleft.org/es/9/que-es-copyleft>.
28. **Parlamento Europeo y Consejo de la Unión Europea.** Directiva 95/46/CE del Parlamento Europeo y del Consejo, de 24 de octubre de 1995, relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos. *Agencia Estatal Boletín Oficial del Estado*. [En línea] 23 / 11 / 1995. [Data: 08 / 08 / 2015.] <http://www.boe.es/buscar/doc.php?id=DOUE-L-1995-81678>. DOUE-L-1995-81678.
29. **Gobierno del Estado Español.** Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. *Boletín Oficial del Estado*. [En línea] 13 / 12 / 1999. [Data: 08 / 08 / 2015.] <http://www.boe.es/buscar/act.php?id=BOE-A-1999-23750>. BOE-A-1999-23750.
30. **Wikipedia.** Web 2.0. *Wikipedia en castellano*. [En línea] 10 de junio de 2013. [Citado el: 12 de junio de 2013.] http://es.wikipedia.org/wiki/Web_2.0.
31. —. Web 3.0. *Wikipedia en castellano*. [En línea] 20 / junio / 2013. [Data: 12 / junio / 2013.] http://es.wikipedia.org/wiki/Web_3.0.
32. —. HTML. *Wikipedia in English*. [Online] junio 19, 2013. [Cited: junio 19, 2013.] <https://en.wikipedia.org/wiki/HTML>.

33. —. Web typography. *Wikipedia in English*. [Online] junio 19, 2013. [Cited: julio 01, 2013.] http://en.wikipedia.org/wiki/Web_typography.
34. **robotstxt.org**. The Web Robots Pages. *About /robots.txt*. [En línea] 2007. [Data: 05 / 08 / 2015.] <http://www.robotstxt.org/robotstxt.html>.
35. **sitemaps.org**. Sitemaps XML format. *sitemaps.org*. [Online] 02 27, 2008. [Cited: 08 05, 2015.] <http://www.sitemaps.org/protocol.html>.
36. **Wikipedia**. Sitemaps. *Wikipedia*. [Online] 07 29, 2015. [Cited: 08 05, 2015.] <https://en.wikipedia.org/wiki/Sitemaps>.
37. —. PDF. *Wikipedia en castellano*. [En línea] 18 / junio / 2013. [Data: 01 / julio / 2013.] <https://es.wikipedia.org/wiki/PDF>.
38. —. ASCII. *Wikipèdia en català*. [En línea] 11 / julio / 2013. [Data: 13 / julio / 2013.] <http://ca.wikipedia.org/wiki/ASCII>.
39. **Ishida, Richard**. Internacionalización. Codificación de caracteres: conceptos básicos. *W3C*. [En línea] 12 / octubre / 2010. [Data: 9 / julio / 2013.] <http://www.w3.org/International/articles/definitions-characters/>.
40. **Wikipedia**. UTF-8. *Wikipedia en castellano*. [En línea] 16 / mayo / 2013. [Data: 13 / julio / 2013.] <http://es.wikipedia.org/wiki/UTF-8>.
41. **Barth, A**. HTTP State Management Mechanism. *Internet Engineering Task Force (IETF)*. [Online] 04 2011. [Cited: 03 11, 2014.] <http://tools.ietf.org/html/rfc6265>. ISSN: 2070-1721.
42. **Wikipedia**. Base de datos. *Wikipedia en castellano*. [En línea] 09 / diciembre / 2013. [Data: 11 / diciembre / 2013.] http://es.wikipedia.org/wiki/Base_de_datos.
43. **ECMA International**. Standard ECMA-262 ECMAScript® Language Specification Edition 5.1 (June 2011). [Online] June 2011. [Cited: diciembre 09, 2013.] <http://www.ecma-international.org/publications/standards/Ecma-262.htm>.
44. **W3C**. Document Object Model (DOM) Technical Reports. [Online] May 02, 2012. [Cited: diciembre 11, 2013.] Document Object Model (DOM) Technical Reports.
45. **Netcraft**. March 2014 Web Server Survey. *Netcraft*. [Online] 03 03, 2014. [Cited: 03 07, 2014.] <http://news.netcraft.com/archives/2014/03/03/march-2014-web-server-survey.html>.
46. **MathJax**. MathJax. [Online] [Cited: 08 16, 2015.] <https://www.mathjax.org/>.
47. **Wikipedia**. MathJax. *Wikipedia*. [Online] 08 15, 2015. [Cited: 08 16, 2015.] <https://en.wikipedia.org/wiki/MathJax>.
48. **Smus, Boris**. Multi-touch Web Development. *HTML5 Rocks*. [Online] 08 21, 2011. [Cited: 09 21, 2015.] <http://www.html5rocks.com/en/mobile/touch/>.
49. **Mozilla Developer Network**. Touch events. *Mozilla Developer Network*. [Online] 09 15, 2015. [Cited: 09 21, 2015.] https://developer.mozilla.org/en-US/docs/Web/API/Touch_events.
50. **W3C**. Touch Events W3C Recommendation 10 October 2013. *W3C*. [En línea] 10 / 10 / 2013. [Data: 21 / 09 / 2015.] <http://www.w3.org/TR/touch-events/>.
51. **Wikipedia**. XML, Extensible Markup Language. *Wikipedia en castellano*. [En línea] 03 / julio / 2013. [Data: 16 / julio / 2013.] http://es.wikipedia.org/wiki/Extensible_Markup_Language.

52. **Bos, Bert.** XML en 10 punts. W3C. [En línia] 13 / noviembre / 2001. [Data: 16 / julio / 2013.] <http://www.w3.org/XML/1999/XML-in-10-points.ca.html>.
53. **W3C.** Web Content Accessibility Guidelines (WCAG) 2.0. W3C. [En línia] W3C, 11 / 12 / 2008. [Data: 04 / 08 / 2015.] <http://www.w3.org/TR/WCAG20/>.
54. **Wikipedia.** Unicode. *Wikipedia en castellano*. [En línia] 14 / mayo / 2013. [Data: 13 / julio / 2013.] <http://es.wikipedia.org/wiki/Unicode>.
55. **Grimaldos, José J.** Tratamiento digital de imágenes. *Tipos de imágenes*. [En línia] 04 / mayo / 2006. [Data: 13 / junio / 2013.] <http://www.grimaldos.es/cursos/imgdig/tipos.html>.
56. **Guyton, Arthur C.** *Tratado de Fisiología Médica*. Séptima. Méjico D.F : Interamericana S.A. McGraw – Hill, 1989. p. 1263. ISBN: 968-25-1520-3.
57. **Urraca-Piñeiro, José Ignacio.** *Tratado de Alumbrado Público*. San Sebastián : Donostiarra, 1988. ISBN 84-7063-155-1.
58. **Wikipedia.** Transport Layer Security. *Wikipedia en castellano*. [En línia] 11 / noviembre / 2013. [Data: 11 / diciembre / 2013.] http://es.wikipedia.org/wiki/Transport_Layer_Security.
59. **Bonals Muntada, Lluís Albert i Ruiz Mansilla, Rafael.** *Transmissió de calor: teoria*. Primera. Barcelona : Aula teòrica, 21. Universitat Politècnica de Catalunya. Iniciativa Digital Politècnica. IDP, 12/1993. p. 191. 978-84-7653-350-5.
60. **González Arribas, Ángel.** Tipos de imágenes y formatos. *APRENDE TIC*. [En línia] [Data: 13 / junio / 2013.] <https://sites.google.com/site/ticvalcarcel/optimizacion-de-imagenes-para-internet/tipos-de-imagenes-y-formatos>.
61. **The Internet Engineering Task Force (IETF).** The application/json Media Type for JavaScript Object Notation (JSON). *Network Working Group*. [Online] The Internet Engineering Task Force (IETF), July 2006. [Cited: diciembre 09, 2013.] <http://www.ietf.org/rfc/rfc4627.txt?number=4627>.
62. **Wikipedia.** TeX. *Vikipèdia*. [En línia] 13 / 07 / 2014. [Data: 16 / 08 / 2015.] <https://ca.wikipedia.org/wiki/TeX>.
63. **Boix i Aragonés, Oriol, Sanz Sapera, Luis i Córcoles López, Felipe.** *Tecnología eléctrica; Electricidad industrial*. Primera. s.l. : Publicacions d'Abast, 01/12/1999. p. 410. Ingeniería eléctrica. 978-84-95355-09-6.
64. **Fernández Salazar, Luis C. i Landa Amezua, Jaime.** *Técnicas y aplicaciones de la iluminación*. Primera. Madrid : McGraw-Hill Interamericana de España S.L., 12/1992. p. 224. Ingeniería mecánica. ISBN 13: 978-84-481-0046-9.
65. **Wikipedia.** SWF. *Wikipedia en castellano*. [En línia] 23 / mayo / 2013. [Data: 18 / junio / 2013.] <http://es.wikipedia.org/wiki/SWF>.
66. —. Standard Generalized Markup Language. *Wikipedia in English*. [Online] junio 05, 2013. [Cited: junio 19, 2013.] <https://en.wikipedia.org/wiki/SGML>.
67. **ECMA International.** Standard ECMA-404 The JSON Data Interchange Format (October 2013). [Online] october 2013. [Cited: diciembre 09, 2013.] <http://www.ecma-international.org/publications/standards/Ecma-404.htm>.
68. **Wikipedia.** SMIL. *Wikipedia en castellano*. [En línia] 26 / abril / 2013. [Data: 17 / junio / 2013.] <https://es.wikipedia.org/wiki/SMIL>.
69. —. Sistema de gestión de contenidos. *Wikipedia*. [En línea] 03 de 09 de 2015. [Citado el: 25 de 09 de 2015.] https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos.
70. —. Scalable Vector Graphics SVG. *Wikipedia en castellano*. [En línia] 11 / junio / 2013. [Data: 17 / junio / 2013.] https://es.wikipedia.org/wiki/Scalable_Vector_Graphics.

71. —. Realidad aumentada. *Wikipedia*. [En línea] 21 de 09 de 2015. [Citado el: 26 de 09 de 2015.] https://es.wikipedia.org/wiki/Realidad_aumentada.
72. **Torrente-Vigil, F. Javier** . Proyecto de fin de máster en sistemas inteligentes. Un marco de referencia para la integración de aprendizaje personalizado basado en contenido altamente interactivo en entornos virtuales de enseñanza. *Universidad Complutense de Madrid*. [En línea] 09 de 2009. [Citado el: 18 de 03 de 2014.] <http://eprints.ucm.es/9984/1/Torrente-PFM09.pdf>.
73. **Wikipedia**. Protocol d'Internet. *Vikipèdia*. [En línia] 20 / 09 / 2014. [Data: 24 / 07 / 2015.] https://ca.wikipedia.org/wiki/Protocol_d'Internet.
74. —. Port (informàtica). *Vikipèdia*. [En línia] 12 / 04 / 2015. [Data: 24 / 07 / 2015.] https://ca.wikipedia.org/wiki/Port_%28inform%C3%A0tica%29#Ports_.l.C3.B2gics.
75. **Bahit, Eugenia**. POO y MVC en PHP. *Programación LAMP con Scrum y XP* . [En línea] [Citado el: 05 de 08 de 2015.] <http://eugeniabahit.blogspot.com.es/p/poo-y-mvc-en-php.html>.
76. **Heurtel, Olivier**. *PHP y MySQL*. [trad.] Daniel Marín. Cornellà de Llobregat : Ediciones ENI, 2009. ISBN: 978-2-7460-4502-6.
77. **Wikipedia**. Optimització per a motors de cerca. *Vikipèdia*. [En línia] 15 / 06 / 2015. [Data: 05 / 08 / 2015.] https://ca.wikipedia.org/wiki/Optimitzaci%C3%B3_per_a_motors_de_cerca.
78. **Oracle**. Object-Oriented Programming Concepts. *The Java™ Tutorials* . [En línea] [Data: 07 / 08 / 2015.] <http://docs.oracle.com/javase/tutorial/java/concepts/object.html>.
79. **Wikipedia**. Nginx. *Wikipedia*. [Online] 03 04, 2014. [Cited: 03 07, 2014.] <http://en.wikipedia.org/wiki/Nginx>.
80. —. Network News Transport Protocol. *Vikipèdia*. [En línia] 29 / 12 / 2014. [Data: 08 / 08 / 2015.] https://ca.wikipedia.org/wiki/Network_News_Transport_Protocol.
81. **Zartner, Sebastian**. MutationObserver. *Mozilla developer network MDN*. [Online] 09 11, 2015. [Cited: 09 06, 2015.] <https://developer.mozilla.org/en-US/docs/Web/API/MutationObserver>.
82. **Edward, James**. Modular Design Patterns: Private, Privileged, and Protected Members in JavaScript. *Sitepoint*. [En línia] 04 / 10 / 2012. [Data: 03 / 09 / 2015.] <http://www.sitepoint.com/modular-design-patterns-in-javascript/>.
83. **Chan, Joanne**. mLearning: The Way of Learning Tomorrow. *eLearning Industry*. [Online] 05 28, 2014. [Cited: 09 27, 2015.] <http://elearningindustry.com/mlearning-the-way-of-learning-tomorrow>.
84. **Wikipedia**. MathML. *Wikipedia*. [Online] 07 30, 2015. [Cited: 08 15, 2015.] <https://en.wikipedia.org/wiki/MathML>.
85. **Westinghouse Electric Co.** *Manual del alumbrado*. Tercera. Madrid : Editorial Dossat, S.A., 1979. p. 256. 978-84-237-0314-2 .
86. **Chapa Carreón, Jorge**. *Manual de instalaciones de alumbrado y fotometría*. [ed.] G. Noriega. Primera. Méjico D.F : Editorial Limusa, SA, 1990. p. 270. Electricidad, fotometría, alumbrado. ISBN 968-18-2972-7.
87. **Philips**. *Manual de alumbrado PHILIPS*. s.l. : Paraninfo, 1984.
88. **Ramírez Vázquez, José**. *Luminotecnia. Colección Enciclopedia Ceac de electricidad*. Octava. Barcelona : Grupo ed. CEAC, Editorial Planeta, 1999. p. 630. Ingeniería mecánica. 978-84-329-6011-6.
89. **Hann, Judith**. *Los amantes de la ciencia*. [trad.] José Luis Díaz de Liaño. Primera. Barcelona : Club Círculo de Lectores, 01/1982. p. 192. 978-84-226-1343-5.

90. **Pritchard, David Christopher.** *Lighting*. Quinta. Harlow : Longman Publishing Group, 1995. p. 272. 9780582234222.
91. **Wikipedia.** Licencia de software. *Wikipedia*. [En línea] 08 / 08 / 2015. [Data: 12 / 08 / 2015.] https://es.wikipedia.org/wiki/Licencia_de_software.
92. **Narboni, Roger.** *La lumière urbaine: éclairer les espaces publics*. Primera. París : Le Moniteur. Collection Technique de Conception, 1995. p. 264. 978-2-281-19088-5.
93. **Resig, John i Bibeault, Bear.** *JavaScript Ninja*. Madrid : Ediciones Anaya Multimedia (Grupo Anaya, S.A.), 2013. 978-84-415-3397-4.
94. **Eckstein, Robert.** Java SE Application Design With MVC. *Java Technology Fundamentals*. [En línea] 27 / 08 / 2008. [Data: 05 / 08 / 2015.] https://blogs.oracle.com/JavaFundamentals/entry/java_se_application_design_with.
95. **Aumaille, Benjamín.** *J2EE. Desarrollo de aplicaciones web*. [trad.] Francesc Garcia. Cornellà de Llobregat : Ediciones ENI, 2003. ISBN:2-7460-1981-7.
96. **Fraile Mora, Jesús.** *Introducción a las instalaciones eléctricas*. Primera. Madrid : UPM. ETSI CAMINOS, CANALES Y PUERTOS, 02/1993. p. 352. Ingeniería mecánica. 978-84-7493-149-5.
97. **Pons, Olga, et al.** *Introducción a las Bases de Datos. El modelo relacional*. Madrid : Thomson Editores. Paraninfo, 2005. ISBN: 978-84-9732-396-3.
98. **Wikipedia.** Internet. *Wikipedia*. [En línea] 15 / 06 / 2015. [Data: 22 / 07 / 2015.] <https://en.wikipedia.org/wiki/Internet>.
99. —. Interfaz de entrada común. *Wikipedia en castellano*. [En línea] 24 / 10 / 2013. [Data: 06 / 03 / 2014.] http://es.wikipedia.org/wiki/Interfaz_de_entrada_com%C3%BAn.
100. **Spitta, Albert F.** *Instalaciones eléctricas*. Primera. Madrid : Editorial Dossat, S.A., 10/1980. p. 830. Vol. I, Ingeniería mecánica. 978-84-237-0417-0 .
101. **Wikipedia.** Imagen de mapa de bits. *Wikipedia en castellano*. [En línea] 11 / marzo / 2013. [Data: 13 / junio / 2013.] http://es.wikipedia.org/wiki/Gr%C3%A1fico_rasterizado.
102. **Aguilar Rico, Mariano i Gimenez, Vicente Blanca.** *Iluminación y Color*. Primera. Valencia : Editorial de la Universitat Politècnica de València, 1995. p. 704. Física-óptica. 978-84-7721-354-3.
103. **Wikipedia.** Hypertext Transfer Protocol. *Wikipedia en castellano*. [En línea] 16 / junio / 2013. [Data: 18 / junio / 2013.] http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol.
104. **php.net.** Historia de PHP. [En línea] [Data: 07 / 03 / 2014.] <http://www.php.net/manual/es/history.php.php>.
105. **Wheater, Paul R., Burkitt, H. G. i Daniels, V. D.** *Histología funcional: texto y atlas en color*. [trad.] Antonio Garfia González. Tercera. Barcelona : Editorial Jims, SA, 11/1987. p. 358. Anatomía. 978-84-7092-296-1 .
106. **W3C.** Guía Breve de Tecnologías XML. *W3C*. [En línea] [Data: 16 / julio / 2013.] <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>.
107. **Wikipedia.** Glifo. *Wikipedia en castellano*. [En línea] 26 / junio / 2013. [Data: 9 / julio / 2013.] <http://es.wikipedia.org/wiki/Glifo>.
108. **Roller, D. i Blum, Ronald.** *Física. Electricidad, magnetismo y óptica*. [trad.] José Aguilar Peris. Barcelona : Editorial Reverté, 1986. p. 580. Vol. 2, Física. 978-84-291-4336-2 .

109. *Estilos de aprendizaje: su influencia para aprender a aprender*. **González-Clavero, Maria Victoria**. 7, 04 / 2011, Journal of Learning Styles. Revista Estilos de Aprendizaje, Vol. 7. http://www.uned.es/revistaestilosdeaprendizaje/numero_7/articulos/lsr_7_articulo_12.pdf. ISSN: 2332-8533.
110. **Folguera Caveda, Eduardo**. Enluminat artificial a l'arquitectura. Barcelona, Catalunya, Espanya : UPC, Escuela Técnica Superior de Arquitectura de Barcelona (ETSAB), Departament de Construccions Arquitectòniques I , 1991.
111. **Fernández Mills, Gonçal i Fernández Ferrer, Julián**. *Electricidad, teoría de circuitos y magnetismo*. Primera. Barcelona : Ediciones UPC, SL, 1996. p. 238. Aula ETSEIB,4. 978-84-89636-86-6.
112. **Gauchat, Juan Diego**. *El gran libro de HTML5, CSS3 y Javascript*. s.l. : Marcombo, 2013. ISBN: 978-84-267-1995-9.
113. **Enriquez Harper, Gilberto**. *El ABC del alumbrado y las instalaciones eléctricas en baja tensión*. [ed.] Grupo Noriega. Primera. Méjico DF. : Editorial Limusa, S.A., 1987. p. 357. Vol. 1, Ingeniería eléctrica. 9681825659.
114. **Comité Español de Iluminación (CEI)**. *Cuadernos de eficiencia energética en iluminación, nº 1. Aplicaciones eficientes de lámparas*. Primera. Madrid : Instituto para la Diversificación y Ahorro de la Energía, 11/1996. p. 132. Vol. 1, Ahorro y eficiencia energética. Alumbrado público.. 978-84-86850-74-6.
115. —. *Cuadernos de eficiencia energética en iluminación nº 2. Aplicaciones eficientes de luminarias*. Primera. Madrid : Instituto para la Diversificación y Ahorro de Energía, 11/1996. p. 100. Vol. 2, Ahorro y eficiencia energética. Alumbrado público. 978-84-86850-75-3.
116. **Meyer, Eric A**. *CSS. Técnicas profesionales para el diseño moderno*. [trad.] Luis Eugenio Ortega Donoso. Primera. Madrid : Ediciones Anaya Multimedia (Grupo Anaya, S.A.), 2011. p. 288. ISBN: 978-84-415-2954-0.
117. **Wikipedia**. Cookie (informática). *Wikipedia en castellano*. [En línea] 07 / diciembre / 2013. [Data: 10 / diciembre / 2013.] http://es.wikipedia.org/wiki/Cookie_%28inform%C3%A1tica%29.
118. **Jiménez Cortés, Carlos**. *Colección Manuales de luminotecnía. Oficinas*. Primera. Barcelona : Grupo ed. CEAC. Editorial Planeta, 10/1997. p. 216. Ingeniería eléctrica. 978-84-329-6051-2.
119. —. *Colección Manuales de luminotecnía. Locales comerciales*. Primera. Barcelona : Grupo ed. CEAC. Editorial Planeta., 05/1998. p. 240. Ingeniería eléctrica. 978-84-329-6052-9 .
120. **TEIDE Editorial**. *Atlas universal geo-económico*. Primera. Barcelona : Editorial Teide, S.A., 05/1996. p. 196. Atlas mundiales/mapas del mundo. 978-84-307-7319-0.
121. **Gray, James**. ASCLMathML: now everyone can type MathML. *MSOR Connections*. [En línea] 10 / 2007. [Data: 16 / 08 / 2015.] <http://journals.heacademy.ac.uk/doi/abs/10.11120/msor.2007.07030026>.
122. **Wikipedia**. Apache HTTP Server. *Wikipedia*. [En línea] 06 / 02 / 2014. [Data: 07 / 03 / 2014.] http://en.wikipedia.org/wiki/Apache_HTTP_Server.
123. —. Analizador de paquetes. *Wikipedia en castellano*. [En línea] 08 / marzo / 2013. [Data: 04 / dicimebre / 2013.] http://es.wikipedia.org/wiki/Analizador_de_paquetes.
124. —. AJAX. [En línea] 19 / diciembre / 2013. [Data: 26 / diciembre / 2013.] <http://es.wikipedia.org/wiki/AJAX>.
125. **López, Julio Gómez**. *Administración de sistemas GNU/Linux*. Primera. Madrid : StarBook Editorial, 2010. p. 308. ISBN: 978-84-92650-46-0.
126. **Wang, Frédéric**. Authoring MathML. *Mozilla Developer Network*. [Online] 06 04, 2014. [Cited: 08 15, 2015.] <https://developer.mozilla.org/en-US/docs/Web/MathML/Authoring>.

127. **JQuery**. Browser Support. [Online] 2013. [Cited: diciembre 18, 2013.] <http://jquery.com/browser-support/>.
128. **Wikipedia**. Capa física. *Vikipèdia*. [En línea] 25 / 09 / 2014. [Data: 24 / 07 / 2015.] https://ca.wikipedia.org/wiki/Capa_f%C3%ADsica.
129. —. Commutació de paquets. *Vikipèdia*. [En línea] 23 / 10 / 2014. [Data: 24 / 07 / 2015.] https://ca.wikipedia.org/wiki/Commutaci%C3%B3_de_paquets.
130. **Burgueño, Pablo F.** . Cómo cumplir la Ley de Cookies. *Blog personal de Pablo Burgueño*. [En línea] 09 de 04 de 2012. [Citado el: 12 de 09 de 2015.] <http://www.pabloburgueno.com/2012/04/como-cumplir-la-ley-de-cookies/>.
131. **Ecured**. Comunicación asíncrona. *Ecured*. [En línea] [Citado el: 24 de 09 de 2015.] http://www.ecured.cu/index.php/Comunicaci%C3%B3n_as%C3%ADncrona.
132. **Wikipedia**. Content management system. *Wikipedia*. [Online] 08 20, 2015. [Cited: 09 26, 2015.] https://en.wikipedia.org/wiki/Content_management_system.
133. **Creative Commons**. Creative Commons. [En línea] Creative Commons. [Citado el: 12 de 08 de 2015.] <http://es.creativecommons.org>.
134. **Wikipedia**. Creative Commons. [En línea] 19 de 06 de 2015. [Citado el: 12 de 08 de 2015.] https://es.wikipedia.org/wiki/Creative_Commons.
135. **Griffiths, Jeff**. DOM MutationObserver – reacting to DOM changes without killing browser performance. *Mozilla hacks.mozilla.org*. [Online] 05 10, 2012. [Cited: 09 06, 2012.] <https://hacks.mozilla.org/2012/05/dom-mutationobserver-reacting-to-dom-changes-without-killing-browser-performance/>.
136. **Benítez, Carlos**. El Patrón Módulo Javascript en Profundidad. *etnasoft*. [En línea] 11 de 04 de 2011. [Citado el: 17 de 08 de 2015.] <http://www.etnasoft.com/2011/04/11/el-patron-de-modulo-en-javascript-en-profundidad/>.
137. *El portafolio del alumno, herramienta estratégica para el aprendizaje*. **Rué, Joan**. 169, 2008, REVISTA AULA. De Innovación Educativa.
138. **Sumper, Andreas, et al.** Experience in remote automation laboratories with virtual and real plants. *CITCEA-UPC*. [Online] 09 28, 2004. [Cited: 09 28, 2015.] International Remote Engineering Virtual Instrumentation Symposium. <http://www.citcea.upc.edu/upload/congres/irevs04a1.pdf>.
139. **Tipler, Paul Allen**. *Física*. Barcelona : Reverté, 1994. Vol. 2. ISBN 10: 8429143696.
140. **Cristea, Flavian**. Grapefruit. *What is Responsive design and how does it help your business*. [Online] 08 2014. [Cited: 07 27, 2015.] <http://www.grapefruit.ro/ideas/what-is-responsive-design/>.
141. **Centro de Referencia en Accesibilidad y Estándares Web**. *Guía de Recomendaciones de Accesibilidad y Calidad Web*. s.l. : Instituto Nacional de Tecnologías de la comunicación (INTECO), 2009.
142. **Observatorio de accesibilidad**. *Guía de Validación de Accesibilidad Web v1.0*. s.l. : Ministerio de Hacienda y Administraciones Públicas. Gobierno de España, 2014.
143. **Agencia española de protección de datos**. Guía sobre el uso de las cookies. *Agencia española de protección de datos*. [En línea] [Citado el: 10 de 09 de 2015.] http://www.agpd.es/portalwebAGPD/canaldocumentacion/publicaciones/common/Guias/Guia_Cookies.pdf.

144. **Ma, Jing and Nickerson, Jeffrey V.** Hands-On, Simulated, and Remote Laboratories: A Comparative. Literature Review. *ACM Digital Library*. [Online] 09 2006. [Cited: 09 28, 2015.] <http://www.stevens-tech.edu/jnickerson/ACMComputingSurveys2006MaNickerson.pdf>.
145. **Wikipedia.** Hashtag. *Wikipedia*. [En línea] 15 de 09 de 2015. [Citado el: 15 de 09 de 2015.] <https://es.wikipedia.org/wiki/Hashtag>.
146. —. Hipermèdia. *Viquipèdia*. [En línia] 02 / desembre / 2013. [Data: 04 / diciembre / 2013.] <http://ca.wikipedia.org/w/index.php?title=Hiperm%C3%A8dia&action=history>.
147. —. Hipertext. *Vikipèdia*. [En línia] 27 / novembre / 2013. [Data: 04 / diciembre / 2013.] <http://ca.wikipedia.org/wiki/Hipertext>.
148. —. HTTP cookie. *Wikipedia*. [Online] 04 10, 2014. [Cited: 03 11, 2014.] http://en.wikipedia.org/wiki/HTTP_cookie.
149. —. Interfaz de usuario. *Wikipedia*. [En línea] 10 de 07 de 2015. [Citado el: 15 de 08 de 2015.] https://es.wikipedia.org/wiki/Interfaz_de_usuario.
150. —. Internet. *Viquipèdia*. [En línia] 09 / 06 / 2015. [Data: 24 / 07 / 2015.] <https://ca.wikipedia.org/wiki/Internet>.
151. —. Internet Information Services. *Wikipedia*. [Online] 02 14, 2014. [Cited: 03 07, 2014.] http://en.wikipedia.org/wiki/Internet_Information_Services.
152. **Defense Advanced Research Projects Agency.** INTERNET PROTOCOL. *The Internet Engineering Task Force (IETF)*. [Online] The Internet Engineering Task Force (IETF), 09 1981. [Cited: 07 24, 2015.] <https://tools.ietf.org/html/rfc791>.
153. **Deering, S., et al.** Internet Protocol, Version 6 (IPv6). *The Internet Engineering Task Force (IETF)*. [Online] The Internet Engineering Task Force (IETF), 12 1998. [Cited: 07 24, 2015.] <https://tools.ietf.org/html/rfc2460>.
154. **Wikipedia.** ISO/IEC 8859-6. *Wikipedia in English*. [Online] mayo 09, 2013. [Cited: julio 13, 2013.] http://en.wikipedia.org/wiki/ISO_8859-6.
155. —. LaTeX. *Vikipèdia*. [En línia] 01 / 02 / 2015. [Data: 16 / 08 / 2015.] <https://ca.wikipedia.org/wiki/LaTeX>.
156. —. Learning styles. *Wikipedia*. [Online] 08 17, 2015. [Cited: 09 21, 2015.] https://en.wikipedia.org/wiki/Learning_styles.
157. Manual de LaTeX. *Wikilibros*. [En línea] 27 de 10 de 2014. [Citado el: 16 de 08 de 2015.] https://es.wikibooks.org/wiki/Manual_de_LaTeX.
158. **Taboada, J.A.** *Manual OSRAM sobre electricidad, luminotecnia y lámparas*. Madrid : Dossat/OSRAM, 1979. ISBN: 84-237-0444-0.
159. **Osmani, Addy.** Learning JavaScript Design Patterns. [Online] 2015. [Cited: 09 03, 2015.] <http://addyosmani.com/resources/essentialjsdesignpatterns/book/>.
160. **Cherry, Ben.** JavaScript Module Pattern: In-Depth. *adequately good. decent programming advice*. [Online] 03 12, 2010. [Cited: 09 03, 2015.] <http://www.adequatelygood.com/JavaScript-Module-Pattern-In-Depth.html>.
161. **w3schools.com.** JavaScript Window - The Browser Object Model. [Online] [Cited: diciembre 11, 2013.] http://www.w3schools.com/js/js_window.asp.
162. **Wikipedia.** Windows-1251. *Wikipedia in English*. [Online] mayo 09, 2013. [Cited: julio 13, 2013.] <http://en.wikipedia.org/wiki/CP1251>.

163. —. Web server. *Wikipedia*. [Online] 02 17, 2014. [Cited: 03 07, 2014.] http://en.wikipedia.org/wiki/Web_server.
164. **WAI-W3C**. Web Accessibility Initiative (WAI). *Web Accessibility Initiative (WAI)*. [Online] W3C, 2014. [Cited: 08 02, 2015.] <http://www.w3.org/WAI/>.
165. **Wikipedia**. Transport layer. *Wikipedia*. [Online] 06 20, 2015. [Cited: 07 24, 2015.] https://en.wikipedia.org/wiki/Transport_layer.
166. **Microsoft Developer Network**. The Browser Object Model. *Microsoft Developer Network*. [Online] Wrox Press Ltd., 1997. [Cited: dicimebre 11, 2013.] <http://msdn.microsoft.com/en-us/library/ms952643.aspx>. ISBN 1-861000-72-3.
167. **php.net**. ¿Qué puede hacer PHP? *php.net*. [En línea] [Citado el: 07 de 03 de 2014.] <http://www.php.net/manual/es/intro-whatcando.php>.
168. **Wikipedia**. Adreça IP. *Vikipèdia*. [En línia] 13 / 05 / 2015. [Data: 24 / 07 / 2015.] https://ca.wikipedia.org/wiki/Adre%C3%A7a_IP.
169. **Rosado, Luís y Herreros, Juan Ramón**. Aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física. *UNED*. [En línea] 2004. [Citado el: 28 de 09 de 2015.] <http://e-spacio.uned.es/fez/eserv/taee:congreso-2004-1003/S1A03.pdf>.
170. **Sintes, Tony**. App server, Web server: What's the difference? *Java World*. [Online] 08 23, 2002. [Cited: 03 07, 2014.] <http://www.javaworld.com/article/2077354/learn-java/app-server-web-server-what-s-the-difference.html>.
171. **W3C**. Architecture of the World Wide Web, Volume One. *W3C*. [Online] 12 15, 2004. [Cited: 03 11, 2014.] <http://www.w3.org/TR/2004/REC-webarch-20041215/>.
172. **Wikipedia**. Archivo de texto. *Wikipedia en castellano*. [En línea] 15 de 07 de 2015. [Citado el: 21 de 07 de 2015.] https://es.wikipedia.org/wiki/Archivo_de_texto.
173. —. Ajax (programació). *Viquipèdia en català*. [En línia] 09 / mayo / 2013. [Data: 26 / diciembre / 2013.] http://ca.wikipedia.org/wiki/Ajax_%28programaci%C3%B3%29.
174. **W3C**. Accessible Rich Internet Applications (WAI-ARIA) 1.0. *W3C*. [Online] 03 20, 2014. [Cited: 08 04, 2015.] <http://www.w3.org/TR/wai-aria/>.
175. —. W3C. *HTML Tags*. [Online] 11 03, 1992. [Cited: 09 26, 2015.] <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>.
176. **Bloom, W. i Fawcett, D. W.** *Tratado de Histología*. 11a. Madrid : Ed. Interamericana.McGraw-Hill, 1995.
177. **Cayless, M.A. i Marsden, Alfred Michael**. *Lamps and Lighting*. [ed.] Arnold. 4th. London : Arnold, 1997. SBN-10: 0340646187.

12 Índice de ilustraciones

Ilustración 1 Laboratorio web remoto	22
Ilustración 2 Laboratorio web virtual.....	22
Ilustración 3 Renderizado del código HTML anterior	28
Ilustración 4 Arquitectura cliente-servidor de dos capas	29
Ilustración 5 Arquitectura cliente-servidor multicapa	29
Ilustración 6 Esquema de la comunicación cliente-servidor.....	30
Ilustración 7 Diseño adaptable a las características del dispositivo	32
Ilustración 8 Matriz de compatibilidad de navegadores web.....	33
Ilustración 9 Actores que participan de la accesibilidad	35
Ilustración 10 Esquema de funcionamiento de la arquitectura MVC.....	38
Ilustración 11 Ejemplo de implementación MVC con un cliente ligero	39
Ilustración 12 Ejemplo de implementación MVC con un cliente pesado y AJAX.....	39
Ilustración 13 Estructura de un objeto en POO	40
Ilustración 14 Imagen original.	57
Ilustración 15 Imagen ampliada donde se aprecia la pérdida de calidad causada por el pixelado.	57
Ilustración 16 Ejemplo de BD relacional	69
Ilustración 17 Comunicación cliente-servidor con AJAX.....	71
Ilustración 18 Funcionamiento del servidor de aplicaciones para PHP	73
Ilustración 19 Captura de pantalla de Google Analytics del tráfico de usuarios del curso (agosto-septiembre de 2015) .	77
Ilustración 20 Mapa web del curso.....	82
Ilustración 21 Mecanismo de publicación de las páginas del curso en la plataforma	84
Ilustración 22 Aspecto final de la página montada.....	91
Ilustración 23 Estructura de carpetas de la plataforma de publicación	91
Ilustración 24 Ventana emergente	98
Ilustración 25 Ejemplo del enunciado de un ejercicio sin javascript	100
Ilustración 26 Aspecto del enunciado con la imagen seleccionada de una lista desplegable	101
Ilustración 27 Botones de selección de la imagen en una explicación	102
Ilustración 28 Ejemplo de ejercicio propuesto	102
Ilustración 29 Ejercicio con resultados desplegados	103
Ilustración 30 Solución del ejercicio	103
Ilustración 31 Paso 1 de la explicación interactiva	105
Ilustración 32 Paso 2 de la explicación interactiva	105
Ilustración 33 Paso 3 de la explicación interactiva	106
Ilustración 34 Paso 4 de la explicación interactiva	106
Ilustración 35 Paso 5 de la explicación interactiva	107
Ilustración 36 Tabla de colores interactiva	108
Ilustración 37 Propuesta de nueva interfaz de usuario para la tabla de colores.....	109
Ilustración 38 Aspecto de la barra de herramientas.....	111
Ilustración 39 Arquitectura básica y funcionamiento de una herramienta	112
Ilustración 40 Ejecución de una rutina en forma secuencial	113
Ilustración 41 Ejecución de una rutina en forma asíncrona	114
Ilustración 42 El problema de las peticiones concurrentes en AJAX	114
Ilustración 43 Esquema de funcionamiento de AJAX y programación multihilo	115
Ilustración 44 Mecanismo de identificación de una sesión de servidor	115
Ilustración 45 Estructura de archivos típica de una herramienta.....	117
Ilustración 46 Relación de los archivos con la arquitectura cliente-servidor	117
Ilustración 47 Diagrama entidad relación de la BD de la aplicación	122

Ilustración 48 Barra de herramientas	124
Ilustración 49 Pantalla de inicio de sesión y acceso a las herramientas	125
Ilustración 50 Pantalla para introducir el correo	125
Ilustración 51 Pantalla para introducir un nuevo usuario	125
Ilustración 52 Esquema de la interfaz de usuario para gestionar las sesiones	126
Ilustración 53 Menú contextual de usuario	126
Ilustración 54 Pantalla para editar el perfil.....	126
Ilustración 55 Interfaz de usuario de la calculadora	132
Ilustración 56 Modo normal de la pantalla.....	132
Ilustración 57 Modo de pantalla con línea de comandos	133
Ilustración 58 Modo de pantalla con mensaje de error.....	133
Ilustración 59 Áreas funcionales del teclado	133
Ilustración 60 Pantalla de ayuda de la calculadora.....	134
Ilustración 61 Secuencia de operaciones de $3 + 5$	135
Ilustración 62 Proceso de cálculo de $3+5\cdot\sin(25+65)$	136
Ilustración 63 Interfaz de usuario de las notas y ejemplo de nota	138
Ilustración 64 Nota.....	139
Ilustración 65 Nota minimizada	139
Ilustración 66 Nota con menú para seleccionar el color.....	139
Ilustración 67 Aspecto del glosario	141
Ilustración 68 Interfaz gráfica del glosario	141
Ilustración 69 Herramienta del resumen	142
Ilustración 70 Aspecto del resumen cuando se muestra el glosario	143
Ilustración 71 Aspecto del menú de la herramienta.....	144
Ilustración 72 Ventana para seleccionar el color de marcado	144
Ilustración 73 Visualización en pantalla.....	145
Ilustración 74 Árbol DOM	145
Ilustración 75 Ejemplo de texto seleccionado	146
Ilustración 76 DOM con rango seleccionado	146
Ilustración 77 Transformación del DOM para un texto sencillo	147
Ilustración 78 Transformación del DOM en un texto que se divide entre varias etiquetas HTML.....	147
Ilustración 79 Aspecto del glosario en el portafolio	149
Ilustración 80 Aspecto de las notas en el portafolio.....	149
Ilustración 81 Aspecto del resumen en el portafolio	149

13 Índice de fragmentos de código

Código 1 Ejemplo de página HTML	28
Código 2 Petición HTTP cliente-servidor	31
Código 3 Respuesta HTTP servidor-cliente	31
Código 4 Ejemplo de implementación de una clase en PHP	41
Código 5 Ejemplo de clase que hereda de la clase rectángulo	41
Código 6 Estructura básica un documento HTML	52
Código 7 Definición del estilo en una etiqueta p	53
Código 8 Ejemplo de uso de atributos ARIA en un menú construido con una lista no ordenada	54
Código 9 Ejemplo de PCRE para validar un DNI/NIE	62
Código 10 Ejemplo de código que trabaja con JSON	63
Código 11 Ejemplo de archivo XML	64
Código 12 Ejemplo de PHP serializado	66
Código 13 Ejemplo de contenido de una <i>cookie</i>	67
Código 14 Fragmento del código de la plantilla	85
Código 15 Ejemplo de código de una página de contenido con los datos de configuración	86
Código 16 Archivo .htaccess	87
Código 17 Archivo index.php	87
Código 18 Porción del código de la clase vista con el constructor y el método que crea la página de salida	88
Código 19 Fragmento de la página con el texto antes de ser procesada	89
Código 20 Fragmento de la página después de ser procesada	90
Código 21 Configuración de la página	92
Código 22 Ejemplo de escritura de fórmulas matemáticas con UTF8	93
Código 23 Ejemplo de escritura de fórmulas matemáticas con entidades HTML	93
Código 24 Ejemplo de código MathML	94
Código 25 Fórmula del flujo del campo eléctrico escrita en MathML	94
Código 26 Ejemplo de fórmula escrita en LaTeX	95
Código 27 Ejemplo de fórmula en notación AsciiMath	95
Código 28 Ejemplo de uso de una solución de servidor con LaTeX para generar fórmulas matemáticas	95
Código 29 Ejemplo de uso de MathJax con notación en LaTeX	96
Código 30 Estructura de un módulo en javascript	97
Código 31 Código HTML que hay añadir a la cabecera del documento	97
Código 32 HTML para crear y usar ventanas emergentes	99
Código 33 Idea para simplificar la notación en el caso de la ventanas emergentes	99
Código 34 HTML para la selección de una imagen con una lista desplegable	101
Código 35 HTML para la selección de una imagen con botones	102
Código 36 Ejemplo de uso de la aplicación que muestra/oculta contenidos de forma selectiva	104
Código 37 Ejemplo de uso de explicaciones secuenciales	107
Código 38 Porció de código de la tabla de colores interactiva	108
Código 39 Archivo worker.js	118
Código 40 Arquitectura del módulo cliente de una herramienta implementado en javascript	119
Código 41 Patrón de programación modular en javascript	120
Código 42 Ejemplo de archivo de la herramienta en el servidor	121
Código 43 Ejemplos de <i>PROCEDURE</i> en MySQL	124
Código 44 Fragmento de la plantilla con el código de las pantallas de la barra	128
Código 45 HTML con el contenido de la barra de herramientas. Obsérvese la etiqueta <div> con la información de los módulos	129
Código 46 Extracto del código HTML de la etiqueta <head> modificado por el <i>script</i>	129

Código 47 Ejemplo de expresión matemática compleja.....	136
Código 48 Expresión matemática en RPN.....	136
Código 49 Código HTML.....	145
Código 50 Marcado en un nodo de texto único	147
Código 51 Marcado en un texto que ocupa diferentes nodos	147

14 Índice de tablas

Tabla 1 Ejemplos de codificación de caracteres	59
Tabla 2 Caracteres ASCII (0-127)	60
Tabla 3 Ejemplo de caracteres ilegibles por problemas de codificación	61
Tabla 4 Comparación entre sistemas de notación de operaciones matemáticas	131
Tabla 5 Ejemplos de ejecución de operaciones matemáticas en notación RPN	131

